UCRL-LR-119748

Large-Eddy Simulation of the Rayleigh-Taylor Instability on a Massively Parallel Computer

Paul A.K. Amala

(Ph.D. Thesis)

March 1995

Lawrence
Livermore
National
Laboratory

DISCLAIMER

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

This report has been reproduced
directly from the best available copy.

Available to DOE and DOE contractors from the
Office of Scientific and Technical Information
P.O. Box 62, Oak Ridge, TN 37831
Prices available from (615) 576-8401, FTS 626-8401

Available to the public from the
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Rd.,
Springfield, VA 22161

Work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract W-7405-ENG-48.

UCRL-LR-119748
Distribution Category UC-405

Large-Eddy Simulation of the Rayleigh-Taylor Instability on a Massively Parallel Computer

Paul A.K. Amala

(Ph.D. Thesis)

Manuscript date: March 1995

LAWRENCE LIVERMORE NATIONAL LABORATORY
University of California • Livermore, California • 94551

Large-Eddy Simulation of the Rayleigh-Taylor Instability on a Massively Parallel Computer

BY

Paul Armas Keith Amala

DISSERTATION

Submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

APPLIED SCIENCE

in the

OFFICE OF GRADUATE STUDIES

of the

UNIVERSITY OF CALIFORNIA

DAVIS

Approved:

Committee in Charge

Large-Eddy Simulation of the Rayleigh-Taylor Instability on a Massively Parallel Computer

Paul Armas Keith Amala
Department of Applied Science
University of California, Davis

December 1994

Dissertation Committee:
Garry H. Rodrigue, Chair (University of California, Davis)
Cecil E. Leith, Jr. (Lawrence Livermore National Laboratory)
Harry A. Dwyer (University of California, Davis)

Abstract

A computational model for the solution of the three-dimensional Navier-Stokes equations is developed. This model includes a turbulence model: a modified Smagorinsky eddy-viscosity with a stochastic backscatter extension. The resultant equations are solved using finite difference techniques: the second-order explicit Lax-Wendroff schemes. This computational model is implemented on a massively parallel computer.

Programming models on massively parallel computers are next studied. It is desired to determine the best programming model for the developed computational model. To this end, three different codes are tested on a current massively parallel computer: the Thinking Machines CM-5. Each code uses a different programming model: one is a data parallel code; the other two are message passing codes. Timing studies are done to determine which method is the fastest. The data parallel approach turns out to be the fastest method on the CM-5 by at least an order of magnitude.

The resultant code is then used to study a current problem of interest to the computational fluid dynamics community. This is the Rayleigh-Taylor instability. The Lax-Wendroff methods handle shocks and sharp interfaces poorly. To this end, the Rayleigh-Taylor linear analysis is modified to include a smoothed interface. The linear growth rate problem is then investigated. Finally, the problem of the randomly perturbed interface is examined. Stochastic backscatter breaks the symmetry of the stationary unstable interface and generates a mixing layer growing at the experimentally observed rate.

Acknowledgments

I Personal Thanks

I would like to offer my deepest thanks to my thesis advisor, Dr. Garry Rodrigue. He taught the classes which gave me my understanding of numerical analysis, computational science, and finite difference methods. He was very patient in advising me on this thesis work, and his guidance was absolutely necessary in the completion of this work.

Chuck Leith was instrumental in interesting me in turbulence modeling, and helped immensely with the nuts and bolts of this thesis work. Twice a week for the past several years he has patiently assisted, advised, and been a mentor for me in ways that I will never be able to repay.

Harry Dwyer taught several great classes on fluid mechanics and computational fluid dynamics which I was fortunate enough to take. His assistance in this, and in reading my thesis is greatly appreciated. I regret that I could not have worked more closely with him than I did.

The Student Policy Committee at Lawrence Livermore National Laboratory made this Ph. D. possible. If it wasn't for their support and financial assistance, I never would have been able to pursue my degree. I would especially like to thank John Anderson and William Chandler - the past and present heads of the Student Policy Committee.

Of course, my parents need to be thanked: Dr. Herbert K. Amala and Audrey E. Amala. They instilled in me a sense of curiosity and the love of learning. They promoted my good qualities and overlooked my bad. They made me into the person I am today; without their love, support, and patience I would never have been able to get this far.

And finally, I dedicate this dissertation to my wife June Vicharinee Amala. Her sense of humor, love, and devotion have been pushed to the limits in the bad times when I was tired and cranky. But she was always there for me. She is the true foundation upon which this dissertation is built.

II Research Support

- This work was supported under the auspices of the U.S. Dept. of Energy by Lawrence Livermore National Laboratory under contract No. W-7405-Eng-48
- This research was performed in part using the resources located at the Advanced Computing Laboratory of Los Alamos National Laboratory, Los Alamos, NM 87545
- The National Center for Atmospheric Research, Boulder CO 80307 which is sponsored by the National Science Foundation.

III About This Dissertation

- This dissertation was written with FrameMaker 4.0 on a Macintosh Centris 650, and printed on an Apple NTX Laserwriter.
- The graphs and plots were developed using the SpyGlass programs Plot (1-D), Transform (2-D), and Dicer (3-D), running on a Macintosh Centris 650.
- The mathematical analysis was performed in part using Maple V R2, on a Macintosh Centris 650.
- The development of the Fortran programs used in this dissertation was done with Language Systems Fortran and MPW, on a Macintosh Centris 650.
- All massively parallel codes used in this research were developed in a series of codes that started on a Cray II, then an Alliant Fx-8, next the BBN TC2000, and finally the Thinking Machine’s CM-5.

Table of Contents

Introduction

1.0 Supercomputers and Programming Models

In the past several years, massively parallel computing has come into its own. The availability of computers which support parallel processing is constantly increasing. So are the choice of the programming models used on these machines.

Through the mid 1980's, the vectorizing supercomputer (such as the Cray 1 and 2) represented the state of the art in computing. Even machines with several processors (such as the Alliant FX-8) used vectorization as a primary means for fast computation. In the latter half of the 1980's however, Thinking Machines (and others such as BBN, NCUBE, etc.) began to build computers based on a different architectural model. These new machines had many processors. While the previous machines where based on processors of the mainframe lineage, these new *massively parallel computers* had processors based on mass-produced microprocessor designs. This gave a cost benefit that allowed a single computer to have hundreds or even thousands of processing units.

With this change in computer architecture came a change in programming models. Previously, the program was thought of as a serial process; now the program had the potential for having many different calculations (not necessarily the same) going on at a given time. The first part of this dissertation will be the investigation of these programming models on a state-of-the-art massively parallel computer, the Thinking Machines Inc. CM-5.

2.0 Computational Fluid Dynamics

These new ideas are then applied to a problem of current interest in computational fluid dynamics. A CM-5 is used to solve the a large-eddy simulation of the Rayleigh-Taylor problem. Large-eddy simulation is one of the methods used in the study of turbulent flows which is currently getting a lot of attention. It is of interest to numerous communities, including the engineering, geophysical fluid flow, and physics communities. This large-eddy simulation is further extended by adding a stochastic backscatter term; this addition is one of the new ideas in the extension of the classical eddy-viscosity models.

With the completed computational code, the classic problem of the Rayleigh-Taylor flow is investigated. The linear growth rate of a single three-dimensional mode is examined. And a problem that has had substantially less attention is also investigated: the growth of random perturbations on an interface.

3.0 Outline of this Research

The outline of the following chapters is as follows:

- Determine the best programming model for the given problem and the given computer. This is done in the chapter “Programming Models and Timing Studies” .
- Solve the Rayleigh-Taylor problem using the selected programming model. This is done in the first half of the chapter “Simulation of the Rayleigh-Taylor Instability” .
- Examine the application of the stochastic backscatter model. This is done in the second half of the chapter “Simulation of the Rayleigh-Taylor Instability” .

Each individual chapter stands on its own. Each has a comprehensive introduction, including relevant references to past and current literature. The results of the research are presented, followed by concluding remarks. All graphs, charts, figures, and tables are included within the text of the chapters.

This dissertation assumes a thorough knowledge of the equations of fluid mechanics, linear Rayleigh-Taylor theory, finite difference methods for partial differential equations, and eddy-viscosity turbulence modeling. The details of these subjects are left out of the text of the individual chapters for brevity’s sake. At the end of the dissertation, there are several detailed appendices. These go into considerable depth on this background knowledge. For an even deeper reading in these topics, there are included references to the current literature, reviews, and comprehensive texts.

Programming Models and Timing Studies

4.0 Introduction

Two broad approaches to parallel programming models can be taken. The first is the SIMD[1] model. For this, each processing element has its own data to work on. However, each processing element carries out the same operation at a given time, following a single stream of instructions. In this case the problem to be worked on must be inherently parallel, or recast in a manner that makes it so. The problem must be able to be divided up into many small pieces each of which can have the same work done on it at the same time. This can be also thought of as a shared memory model in that all data are globally accessible by any processor. The early models of the Connection Machine (such as the CM-2 and CM-200) are examples of the SIMD approach.

The second broad approach is that of the MIMD[2] model. In this case each processing element not only has its own data; it also executes its own instructions. This is more broadly parallel in that many different computations can be going on at the same time. This can also be thought of as a distributed memory model; each processor has its own local memory which only it can access directly. The BBN TC 2000 is an example of a computer used in the MIMD mode.

The newest version of the Connection Machine, the CM-5, also allows both programming models [1] [2]. The CM-5 has up to 1024 processors which may be divided into partitions each controlled by a front end computer (e.g., a Sun SPARC station.) Each processor has its own local memory. A parallel version of Fortran (CM Fortran) or C (named C*) can be used to program the machine in a SIMD mode. Or message passing libraries (CMMD) can be used to allow the machine to function in the MIMD mode. A nice feature is that the partition does not have to be restarted to change modes; in fact, when the CM-5 is being used

[1] Single Instruction, Multiple Data

[2] Multiple Instruction, Multiple Data

in a time sharing manner, different users can be using the different methods simultaneously.

Both approaches have their advantages and disadvantages; the best choice depends more on the problem being solved than on the individual computer in question. To further complicate the matter, compiler technology has often not caught up with the machines they are being used upon. Unless the programmer is going to optimize his code by machine assembly language coding of time intensive subroutines, [11] [12], the applications programmer is at the mercy of the machine's vendor to supply a compiler for a given language and model that "does the right thing."

There are numerous articles and text on parallel processing and computer architecture. See [44] or [46] for example. A good background reference to other problems that have been solved on the CM-5 is given in [41]. Long [67][68] gives timing results for a 3-D Navier-Stokes solver with various parallel computers, including the CM-2 (the predecessor to the CM-5.)

4.1 Purpose of this research

This portion of the dissertation looks at three different approaches to programming massively parallel computers. Specifically, the CM-5 at the Advanced Computing Laboratory (ACL) at Los Alamos National Laboratory is used to perform timing runs on a problem from computational fluid dynamics. The ACL CM-5 is a 1024 processor machine, usually divided into partitions of 32, 64, 128 and 512 processing elements which operate as independent machines. Each processing element has a scalar SPARC chip[3] and four associated vector units (VU) (Figure 1.) Each partition can be thought of as a separate parallel computer, independent of the other partitions.

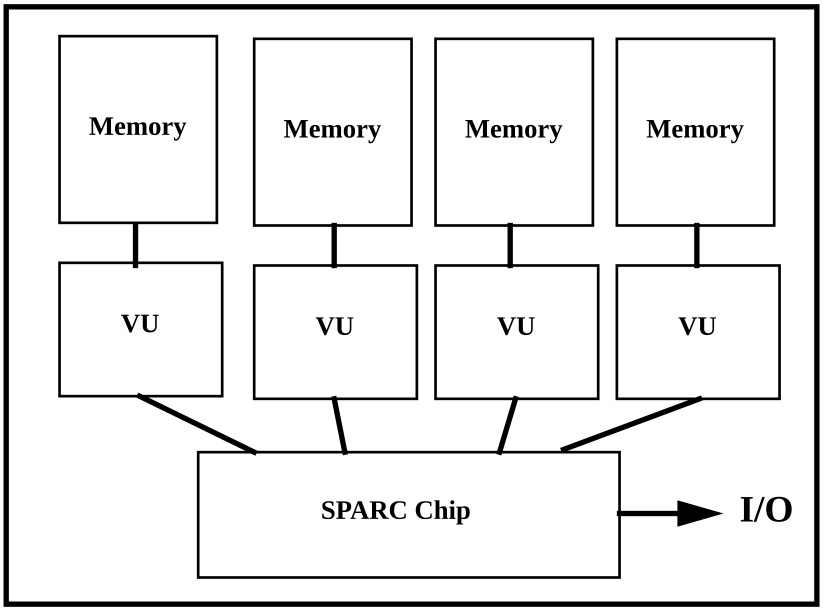
4.1.1 The first approach

The first approach is to solve the problem using data parallel Fortran (CM Fortran or CMF.) This is a SIMD approach that fully abstracts away any data transfer questions from

[3] Sun Microsystems Inc. RISC microprocessor

the programmer. The compiler handles any data movement between processing elements. From the point of view of the program all memory is shared. This method can fully take advantage of the vector units (VU) recently installed on the ACL CM-5.

FIGURE 1. A CM-5 processing node



4.1.2 The second approach

The second approach uses CM Fortran restricted to each node. Each CM-5 node has four vector units (VU;) this approach treats each node as a parallel computer of four processors. Message passing (via the CMMD package) is explicitly used to transfer data between nodes, while internal (i.e., between the four VU) communications is handled by the CMF compiler. This is a MIMD approach to the computational problem, where data is distributed between the various nodes; no node has access to another’s data except via message passing.

4.1.3 The third approach

The final approach is a remnant of the days before the CM-5 had the VU installed. Each node is treated simply as a scalar SPARC chip. Fortran 77 is used to program serially each node, and the CMMD library package is used for explicit message passing of boundary values between nodes.This method of implementing message passing is still used on a number of current massively parallel machines, including the Paragon, the Meiko, and the Touchstone Delta. Because it uses Fortran 77 (which is available on most machines) and a library of common message passing calls (usually implemented as a macro package or Fortran library) this method is highly portable.

These three approaches are applied here to a numerical method of current interest in computational fluid dynamics. Timing runs are done for a simple test problem, and the results are analyzed in an attempt to single out the best approach for this type of computational problem, given current software and hardware implementations for the CM-5. Care is taken so that the three codes do the same type and amount of work. However, the most efficient techniques are used within each of the three methods so that the times represent the best that can be achieved with the given programming model.

5.0 The Computational Problem

Often timing studies are done on simple problems. This will achieve very optimal results showing the computer to be very fast. This study takes the opposite approach. A problem of current interest will be used to perform the timing study. This problem is from computational fluid dynamics: a large-eddy simulation code. This problem has everything: the numerical solution of coupled non-linear partial differential equations; the need for a random number generator; a large number of subroutine calls; message passing in two or three different dimensions; and the need to call subroutines from a “canned” mathematical library package to name a few. Instead of being highly optimized to run fast, the code tested is a production code used to solve real world fluid dynamics (such as the Rayleigh-Taylor problem, as will be seen in the next chapter.)

5.1 The Navier-Stokes Equations and Constitutive Relationships

The three-dimensional Navier-Stokes equations are used to model the fluid flow [101]. The fluid is assumed to be an isotropic Newtonian fluid in which the Stokes hypothesis holds. The fluid obeys the Fourier heat law, and is an ideal perfect gas (i.e., a gamma-gas law holds.) The equations, constitutive relationships, and non-dimensional constants are fully described in “Appendix A - The Equations of Fluid Flow” .

Furthermore, the fluid equations are modified to include a turbulence model of current interest to allow for future study of turbulent flows [19][34][61]. The Smagorinsky eddy-viscosity model is used [97]. This is further modified by adding a stochastic backscatter term to account for the effect of subgrid-scale interactions with the larger resolved scales[58][59][72][73]. The details of this turbulence model as used here are set forth in “Appendix B - Turbulence Modelling” .

5.2 The physical problem and boundary conditions

For the timing runs the physical domain is assumed to be periodic in all three dimensions. There are no external body forces (such a gravity,) nor are there any external heat sources. The gas is assumed to be a compressible ideal gas with molecular viscosity insignificant as compared to the eddy-viscosity.

Initial conditions are set from the travelling wave solution to the linearized one-dimensional gas dynamics equations. This solution is (for waves travelling in the x-axis direction):

$$\begin{aligned} \rho &= \rho_0 (1 + \varepsilon \sin(kx - kct)) \\ \rho u &= \varepsilon \rho_0 c \sin(kx - kct) \\ \rho e &= \frac{c^2 \rho_0}{\gamma - 1}\left(\frac{1}{\gamma} + \varepsilon \sin(kx - kct)\right) \end{aligned} \qquad \text{(EQ 1)}$$

This problem was chosen for several reasons. The first is that an analytic solution allows for checking the accuracy of the code and numerical scheme. The second is the ease of generating initial conditions. A third reason is the smoothness of the solutions. This allows the code for the turbulence model to be tested without overloading it (after all, the goal here is to check the timings of the three versions of the code - examination of the turbulence model is the work of a subsequent chapter.)

5.3 The numerical scheme

The five resultant coupled non-linear partial differential equations are solved with the two-step MacCormack scheme. This is an explicit, second-order accurate finite difference method, usually classified as a Lax-Wendroff type scheme. The particulars of the numerical scheme are fully laid out in “Appendix C - Finite Difference Methods” .

6.0 Parallel Fortran (CM Fortran)

Connection Machine Fortran (CMF) is based on Fortran-90, [4]. The main feature of CMF for the problem studied here is the use of array syntax. This feature allows for simplified coding by removing the necessity of indexing arrays. For instance, adding two matrices in Fortran 77 is commonly done with the loop:

```
      do 2000, i = 1, 100
       do 1000, j = 1, 100
         c(i,j) = a(i,j) + b(i,j)
1000 continue
2000 continue
```

In CMF this is replaced by one line:

```
c = a + b
```

With array syntax, a vector v can be accessed from a matrix A with a command such as (a gather type operation):

```
v = A(3,:)
```

Here, the desired vector is the third row of the matrix A, and the number of elements in v is the same as the number of column elements of A.

Each computational point (i, j, k) is placed on a virtual processor (VP.) A virtual processor can be regarded as a small processor which handles only one grid point of data, and is in communication with its neighbors. The topological layout of these virtual processors is handled by the complier. However, it is possible for the programmer to help the compiler out by using the *cmf$ layout* compiler directive. If an array's dimension is specified as *:news*, each of that dimensions grid points will be laid out on a different VP. If a dimension is specified as *:serial*, the dimension will be laid out entirely on one actual processing node. Thus if one assigned a directive of

```
cmf$ layout A(:news,:news)
```

to the matrix A, each grid point A(i,j) would be placed on a separate virtual processor. The run time environment and the compiler determine the actual processing node this virtual processor is assigned to. This is all done at run time.

On the other hand, by applying the directive

```
cmf$ layout A(:serial,:news)
```

to the matrix A, each row of A would be on the same VP. This will guarantee that at run time each entire row of A would reside on only one actual processing node. The various rows would be laid out across the available nodes in a manner determined by the compiler and run time environment.

There are several advantages to using the shared memory/SIMD approach with parallel Fortran. First, the programmer does not have to be concerned with the topology of the processing nodes. The compiler will determine a reasonable arrangement (with a little help from the programmer sometimes using the aforementioned directives.) Another advantage is that the program can be run on partitions of different sizes without having to be recompiled. The run time environment takes the VP layout generated by the compiler, and spreads it onto all the nodes of the partition. A third advantage is that all nodes within the partition are used, with essentially the same number of VPs assigned per actual node. This helps in load balancing and efficiency (assuming that the problem does not have some locations which are computationally intensive.) A final advantage is that the program is very clean. It is easy to see the mathematics in the code with all the do loops and array indexing removed. This aids in debugging since the code is much more readable.

A major disadvantage of CMF is its lack of portability. CMF only works on the Thinking Machines Corp. line of Connection Machines. While High Performance Fortran (HPF) and Fortran-90 are becoming standards, they are currently implemented on few machines.

6.1 Execution time results

Timing runs were made on partitions of various sizes including 32, 64, and 128. The grid sizes used in the computational problem included 40^3 , 64^3 , and 128^3 .

There are two different ways to run a code using parallel Fortran. First, one does not use the vector units. In this case a 32 node partition behaves as if it were 32 SPARC processing units (see Table 3 .) The second way is to use the vector units (see Table 1 .) Each node in the partition has 4 vector units; a 32 node partition behaves as if it were a 128 node vector processing machine. Each vector unit has its own memory (in fact, when using the CM-5 without the vector units for arithmetic operations, the SPARC chip still accesses the node's memory through the vector units.)

Runs were done with and without using the vector units (a compile time option.) Remember that for each processing node there are four vector units. However, the vector units and the SPARC chips are not identical; they have different clock speeds and architectures. Thus it is not correct to state that four vector units will be four times faster than one SPARC chip; in fact the VU are more powerful than a SPARC chip so that the speed-up should be greater than four. The exact scaling will be seen below.

All codes were run with a fixed 24 time steps, a fixed grid element size, and a fixed CFL condition. Execution times are measured for these 24 time steps, and the time in seconds to perform one step is presented in the tables. Thus *Time/Step* is the amount of time spent to perform one complete MacCormack time integration - both the predictor and corrector steps. Time to write states to disk, initialization times, and random number generation timings are not included in the time count (differences between the three code versions do not allow direct comparisons.) *Size* refers to the number of grid points in each dimension, i.e, 64 refers to a total problem size of 64^3. *Nodes* refers to the number of processing nodes allocated at run time; it is the size of the partition the code is run upon. *Mflops/Node* is the estimate of the number of floating point operations (in units of a million) that are being computed on a single node. *Total Mflops* is the estimate of the overall performance of the partition; it is the product of Mflops/Node and Nodes.

TABLE 1. CMF with vector units

Size	Nodes	Time/Step	Mflops/Node	Total Mflops
40	32	0.130	17.89	572.61
40	64	0.085	13.68	875.35
64	32	0.438	21.74	695.66
64	64	0.228	20.88	1336
128	128	0.855	22.26	2849.3

There are certain things to notice about these results. The first is that doubling the number of processing nodes (for a given problem size) does seem to double the speed of execution (well, almost.) Another is that the CM Fortran's compile time optimizer (using the -O flag) does not seem to help (Table 2 .) In fact, it slightly slows things down (except in one case.)

TABLE 2. CMF with vector units, -O optimized

Size	Nodes	Time/Step	Mflops/Node	Total Mflops
40	32	0.133	17.51	560.39
40	64	0.084	13.85	886.21
64	32	0.509	18.70	598.35
64	64	0.262	18.13	1160.6
128	128	1.014	18.77	2402.5

The non-vector unit runs were done under the same conditions as above.

TABLE 3. CMF without vector units

Size	Nodes	Time/Step	Mflops/Node	Total Mflops
40	32	1.264	1.839	58.851
40	64	0.625	1.861	119.08
64	32	8.324	1.144	36.597
64	64	3.919	1.214	77.718
128	128	17.642	1.079	138.13

It is clear that there is a speed-up from the use of the four VU. The speed-up factor for the above runs goes from a low of just over 7 to a high of almost 21. This is greater than the potential speed-up gained by a simple increase from one to four SPARC chips. Thus the VU are giving more than just a scaling increase due to increasing the number of processing elements.

6.2 Observations

In its initial form the program seemed to spend most of its time in passing boundary values. Later it was realized that the use of ghost cells[4] to enforce periodic boundary conditions was quite inefficient. In CMF, the *CSHIFT* command is capable of establishing periodic boundary conditions without the artifice of ghost cells. This saves in memory usage; also the method of moving data between processing nodes is implemented more quickly than the array partitioning commands necessary to set up periodic boundary conditions explicitly with ghost cells.

Doubling the number of processors does not quite double the speed of the calculation for a fixed problem size. This can be accounted for by an increase in the amount of communication between nodes that must be done (implicitly via the CSHIFT command.) This is a simple surface to volume problem; as the number of processing nodes increases for a fixed problems size, more surface area (inter-nodal boundaries) exists. This implies the need for an increase in inter-nodal communication which is expensive. The lesson here is that it pays to have the node do as much work as possible with as little communication as possible.

The speed of these runs is quite impressive. Rough scaling arguments show that the 128^3 problem would run on a 1024 node partition at a total rate of approximately 20 GFlops. This is very respectable indeed.

[4] i.e., grid points inserted beyond the computed domain for boundary value evaluation

7.0 Parallel Fortran with Message Passing

In the previous section CMF was used in a data parallel programming model. CMF can also be used in another way. If the vector units (VU) are present, each node can be thought of as a 4 processor parallel node. CMF will execute on each node of 4 VU, using message passing (via the CMMD library routines) to pass boundary values between nodes, [9], [10].

In this case, it is possible to have different nodes doing different tasks (each node has a unique identification number and thus the program can assign tasks based on this.) It is also possible to have the front end computer acting as the host for the program, although this is not a recommended mode for the use of the machine. Thinking Machines Corp. recommends that the *hostless* model be used instead, [9]. In the hostless model, the code only executes on the CM-5. There is no front end computer coordinating the execution of the CM-5 code (other than the ever present I/O handlers, loaders, time-sharing managers, etc.)

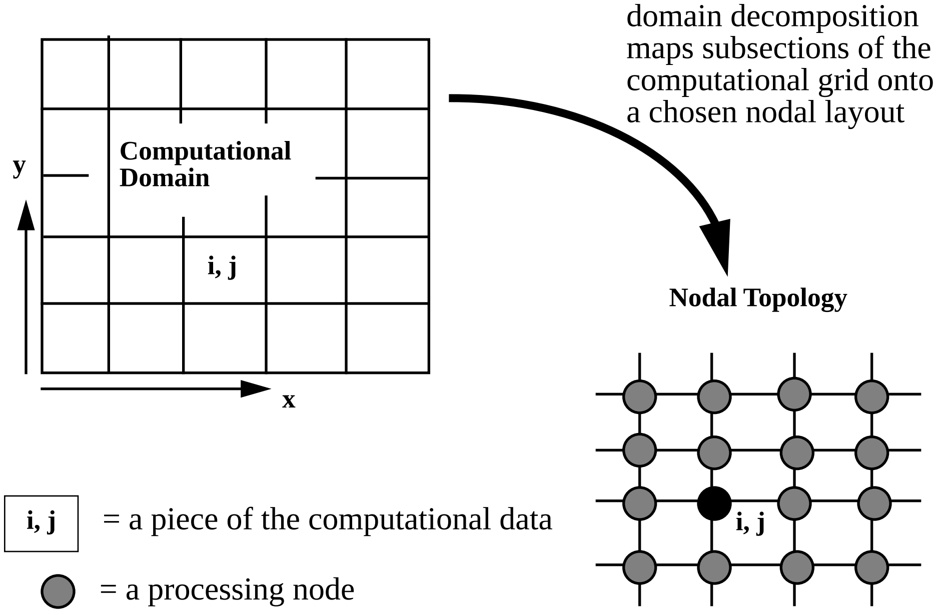
The code tested here is a *hostless* code. Node 0 acts as a master, and nodes 1 - N act as slaves carrying out the computation. Because partition sizes must be 32, 64, 128, etc. it is also possible that some nodes will do nothing due to the geometric constraints of the problem. These are waste nodes, and the code they execute must explicitly have them idling. This is a source of inefficiency however; the previous use of globally parallel CMF used all nodes in the partition while this use of message passing has the possibility of wasted processor cycles.

This method does have its advantages however. In the globally parallel case, the problem had to be entirely granular. That is, since every processor must execute the same code there is no ability for different things to be going on at the same time. With message passing several processors can be handling I/O, many processors can be working on the computation at hand, and a few more can be working out diagnostics or values used in graphics. And this can all go on at the same time. This is not possible with global parallel Fortran where one instruction is executed by all the nodes at the same time.

7.1 Domain decomposition

The concept of domain decomposition is used in this approach. Here, the computational domain is divided up into smaller portions and each of these portions is given to a separate processing node[5] (see Figure 2.) However, while the physical problem only has its exterior boundaries to worry about, each of these sub-domains will now have its own boundaries too. These boundaries are artificial (except for those that also correspond to actual physical exterior boundaries) - hence the name *ghost cells (boundaries.)* Their values are still computed explicitly on the processing node of a neighboring subdomain. Then message passing is used to transmit the actual values from the node where they are computed into the ghost cells of the node where they will be used as boundary conditions (see Figure 3.)

FIGURE 2. Domain decomposition



domain decomposition maps subsections of the computational grid onto a chosen nodal layout

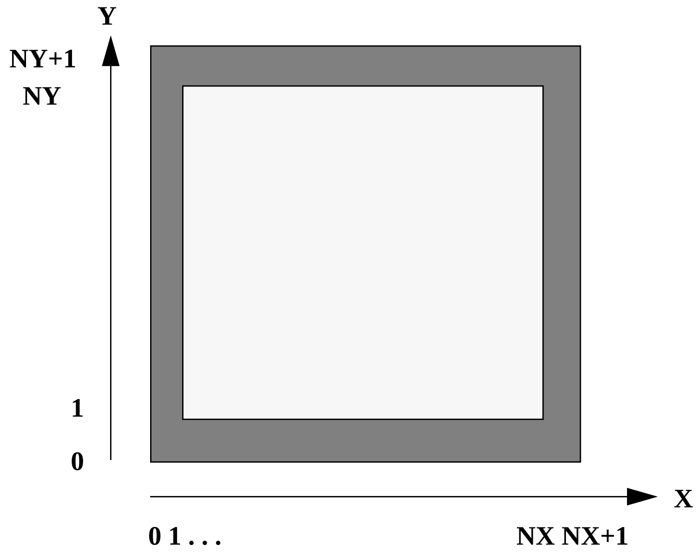
Hence, in the concept of domain decomposition each node has its own data set to work upon. Since each node is made of 4 VU, the CMF compiler will further divide up this data on a node among the 4 VU. The nodal data set must also include boundary information; this boundary data is computed on a neighboring node. Likewise, some of the actual data

[5] typically called a *slave node*, since it does the work

computed on this node is needed by its neighbor for their boundary values. These boundary values are needed to complete the finite difference formulas at the boundaries. Thus, each node is a separate computer with its own data; the CMMD library handles the intercomputer communications.

It is also possible not to use the VU on the nodes. Then the CMF compiler views each node as a computer with one processor: the scalar SPARC chip. However, the message passing strategy outlined above is still used for the inter-nodal communications.

FIGURE 3. Ghost cells



□ = interior grid points computed on the node

■ = exterior grid points received via message passing

It is important to note that in order to use the VU on a node, the CMF compiler must be used. This is the only approach possible. Fortran 77 on a node with message passing will not allow the use of the VU (except for some special libraries that can be loaded at link time.) Fortran 77 on a node will be discussed in Section 8.0 on page 24.

7.2 Ghost cells for handling subdomain boundaries

The process of moving data to/from neighboring ghost cells is important and worthy of elaboration. The interior data points are computed on the node in question; the exterior (2-D surface if the data set is 3-D) points must come from the neighboring nodes where they are computed as interior points. This entails two steps within the message passing model. First a 2-D data structure must be stripped from a 3-D data array (a *gather* operation called *sectioning*.) Then, this 2-D structure must be sent to its proper neighbor using the CMMD message passing library. For example:

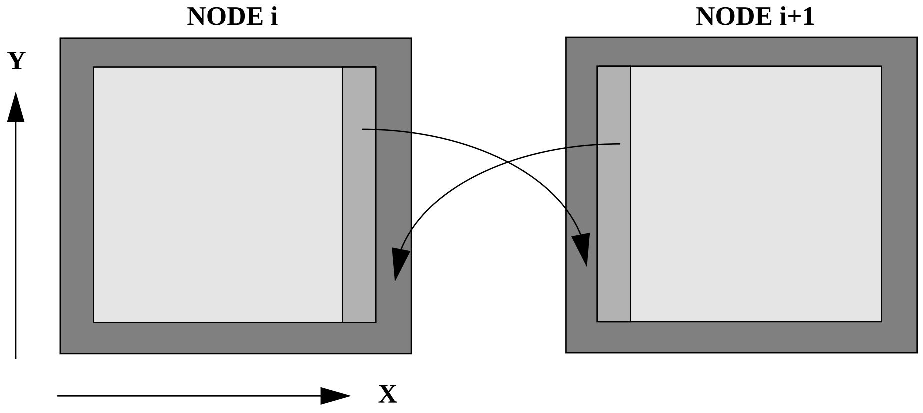
$$\begin{gathered} Surface(i, j) = Volume(i, j, NZ) \\ Surface_n \rightarrow Surface_{n+1} \end{gathered} \qquad \text{(EQ 2)}$$

The opposite must be done to get the node's own boundary values. The 2-D message is received from the proper neighboring node, and then it is *scattered* into the 3-D data structure. For example:

$$\begin{gathered} Surface_{n+1} \rightarrow Surface_n \\ Volume(i, j, NZ+1) = Surface(i, j) \end{gathered} \qquad \text{(EQ 3)}$$

Here the top z-layer computed on node n is sent to node n+1. The 2-D case is illustrated in Figure 4.

FIGURE 4. Message passing of ghost cell boundary values



An internally computed right column of Node i is sent as a message to become the left boundary of Node i+1. Likewise, the internally computed left column of Node i+1 is sent to Node i to become its right most boundary.

7.3 Comparison considerations

There are two ways to think of comparing timing with the above global CM Fortran method. The first would be to look at it from an "economical" point of view. For a given machine which has, say, 32 or 64 nodes, what can be done? Thus a CMF code using all 32 nodes in a global sense is directly compared to a CMF/CMMD code using 1 master, 25 slave and 6 waste nodes. However, it must be noted that the CMF version will be at an advantage; in the main loops it is getting all 32/32 of the processors to work on the data set. The CMF/CMMD version is only getting 25/32 of the processors for the same size data set.[6]

Another method of comparison would be to compare the 32 nodes that work on the CMF data set to 32 nodes of working effort for the CMF/CMMD version. However, given the fixed size of partitions this would mean that the CMF/CMMD code would actually need a partition of size 64. Then it could allocate 1 master node, 32 slave nodes to do the actual work, and 31 waste nodes. This is, of course, a terribly inefficient way to allocate resources. It is only described here as a second means of comparison.

7.4 Execution time results

Timing runs were made on partitions of various sizes including 32, 64, and 128. The grid sizes used in the computational problem included 40^3 and 64^3.

An important consideration is layout. On the global CMF code, the compiler (except for *layout* compiler directives) determines how to place the data on the processing nodes. In the CMF/CMMD code considered in this chapter, the programmer makes these decisions. Thus it is possible that laying out x- and y-axes across the nodes and all of a z-axis (for a fixed value of x and y) on one node may behave differently that another layout. There is also the question of nodal topology. Here, a two-dimensional layout of the nodes is used. This is done in order to have as few waste nodes as possible; there are more perfect square numbers in 32 or 64 than there are cubes. All in all, there are many possible mappings of a

[6] Actually, taking into account the ghost cells, the CMF/CMMD version also works on a larger data set for the same amount of actual work.

three-dimensional problem onto a given number of nodes. Two different geometries are examined here.

The first geometry is to lay out the entire x-axis (for a given value of y and z) upon on a node. Then the yz-plane will be laid out across different processing elements. To ensure that a dimension of an array is placed entirely on a node, it is declared *:serial* (as opposed to *:news* which allows the compiler to figure out how to place it - usually across nodes.) This will ensure that within a processing node, an entire x-axis is placed on only one VU. Thus in our mapping the x-axis is a "long" axis, and the y- and z-axes are broken up using the aforementioned domain decomposition techniques.

Another possible geometry is to let the compiler figure out how to lay out the decomposition across the nodes. All spatial dimensions are declared *:news*, with the further constraint that the z-axis is the "long" axis. This corresponds to how the previous parallel Fortran version was run. The compiler must figure out the proper way to parcel out the subdomain among the 4 VU on a node.

Within these constraints, there are two compiler directives related to the layout across nodes. The first uses the *cmf$ layout* compiler directive. This allows the compiler to lay out the computational elements across virtual processors (VP) in the manner it determines to be most efficient. In general, conformal arrays will be placed on the same VP. The second directive uses the *cmf$ align* compiler directive. This allows the programmer to force the compiler to align a given array with another array already laid out with the *cmf$ layout* directive (i.e., this guarantees that the two arrays with be placed on the same VPs,) see [1], [5] for more details.

Runs with these various nodal layouts were done to observe their effect upon run times (and hence the efficiency of the compiler/CMMD libraries.) The run time memory requirements were excessive. The 64^3 problem would not execute on a partition size of 32 nodes; the partition memory limit is 1 Gigabit of memory. Note that the global CMF version did not have any problems executing this problem within the 32 node/1 G limits.

The efficiency is quite low from another perspective. In most of the runs, it was only possible to use 50% of the allocated nodes on the computation. This is within the limits of having each subdomain being the same size. The subdomains could be set up to be of non-uniform size to permit the use of more of the allocated nodes. However, this may additionally introduce load balancing problems.

In the tables below, the layout of nodes and subdomains is listed in the format *number of blocks:size of blocks*.Thus B:NX = 5:8 refers to a layout that has 5 rows of nodes in the x-direction, each of which is 8 elements wide. Furthermore, if B:NY = 5:8 and B:NZ = 1:40, the total problem size would be 40^3 with 25 slave nodes laid out over the xy-plane, with the z-axis being the "long" one. Since the total size of the partition must be a power of 2, this problem would have to be run on a 32 node partition, with a one node being the master, and the other 6 nodes being wasted.

The column *% Comm.* gives the efficiency of the inter-nodal communications using the CMMD library calls. This is the percent of time in a given time step which is used to communicate the boundary values to neighboring ghost cells. This time does not include the CSHIFT which is intra-nodal, i.e, between VU within a given processing node. *Comm. Time/Word* is the amount of time in microseconds used to send/receive a four byte word of data using the CMMD command cmmd_send_and_receive.

TABLE 4. CMF/CMMD with separated gather/scatter, Z-Axis long (cmf$ align)

B:NX	B:NY	B:NZ	Time/Step	% Comm.	Mflops/ Node	Total Mflops	Comm. Time/word
5:8	5:8	1:40	1.428	81.30%	2.141	53.516	11.822
4:10	8:5	1:40	1.323	83.34%	1.814	58.047	11.113
8:5	4:10	1:40	1.278	82.38%	1.878	60.089	10.122
5:8	8:5	1:40	1.036	82.13%	1.858	74.338	9.724
8:5	5:8	1:40	1.028	83.22%	1.874	74.956	9.717
8:5	8:5	1:40	0.785	81.13%	1.548	99.056	10.349
4:16	8:8	1:64	4.050	82.35%	2.396	76.675	15.834
8:8	4:16	1:64	3.743	80.66%	2.592	82.958	14.219
8:8	8:8	1:64	2.253	81.98%	2.168	138.760	12.223

Table 4 represents the best results that were achieved using CMF/CMMD. These runs were done with a primary array laid out using the *cmf$ layout* directive, and all sub-arrays determined by the *cmf$ align* directive. The timings in Table 5 are identical, except all arrays are laid out using the *cmf$ layout* directive. In both sets, a separate gather is done for an outbound message as is a separate scatter for the data of the inbound message. This is as opposed to using array syntax sectioning to perform implicit gather and scatters within the cmmd_send_and_receive instruction.

TABLE 5. CMF/CMMD with separated gather/scatter, Z-Axis long (cmf$ layout)

B:NX	B:NY	B:NZ	Time/ Step	% Comm.	Mflops/ Node	Total Mflops	Comm. Time/word
5:8	5:8	1:40	2.319	90.95%	1.318	32.95	1.452
4:10	8:5	1:40	2.010	91.35%	1.194	38.22	1.418
8:5	4:10	1:40	2.007	91.33%	1.196	38.269	1.315
5:8	8:5	1:40	1.693	91.89%	1.137	45.498	1.366
8:5	5:8	1:40	1.690	91.91%	1.139	45.573	1.330
8:5	8:5	1:40	1.303	91.89%	0.932	59.635	1.523
4:16	8:8	1:64	6.159	90.78%	1.576	50.417	1.405
8:8	4:16	1:64	6.143	90.87%	1.580	50.549	1.253
8:8	8:8	1:64	3.518	91.19%	1.388	88.858	1.376

Notice that the first set is about 1.5 times faster than the second. This indicates that explicitly aligning arrays improves performance, as opposed to allowing the compiler to do this. However, also notice that the communication time is increased by almost a factor of ten! Apparently, the compiler (via the *cmf$ layout* directive) tries to align the arrays to optimize nodal communications, as opposed to *gather* type operations. A further examination of this topic will be given in the following section.

TABLE 6. CMF/CMMD with X-Axis long (:serial,:news,:news)

B:NX	B:NY	B:NZ	Time/Step	Mflops/ Node	Total Mflops
1:40	5:8	5:8	5.460	0.560	13.997
1:40	4:10	8:5	3.443	0.697	22.308
1:40	8:5	4:10	3.410	0.704	38.269
1:40	5:8	8:5	3.397	0.567	22.679
1:40	8:5	5:8	3.388	0.569	22.743
1:40	8:5	8:5	2.243	0.542	34.667
1:64	4:16	8:8	15.273	0.635	20.331
1:64	8:8	4:16	15.361	0.632	20.214
1:64	8:8	8:8	8.612	0.567	36.302

Table 7 and Table 6 use the array syntax for the implicit gather/scatters necessary to separate a 2-D data structure for message passing. The first is with the x-axis “long”. Here the x-axis is forced to be placed entirely upon a single VU for a given value of y and z.

The final set of runs are the z-axis “long” set. While the programmer constrains the topology of the nodal layout, the compiler is allowed to lay out the subdomains among the 4 VU in any way it desires. Since both of these use the *cmf$ layout* directive, they are directly comparable to Table 5 . Notice that the both of these are slower that than Table 5 . From this we conclude that both nodal topology, and the use of separated gather/scatter operations (as opposed to implicit use) is important in overall performance.

TABLE 7. CMF/CMMD with Z-Axis long (:news,:news,:news)

B:NX	B:NY	B:NZ	Time/Step	Mflops/ Node	Total Mflops
5:8	5:8	1:40	2.997	1.020	25.503
4:10	8:5	1:40	2.629	0.913	29.214
8:5	4:10	1:40	2.625	0.914	29.263
5:8	8:5	1:40	2.208	0.872	34.889
8:5	5:8	1:40	2.204	0.874	34.955
8:5	8:5	1:40	1.711	0.710	45.453
4:16	8:8	1:64	8.256	1.175	37.613
8:8	4:16	1:64	8.248	1.176	37.647
8:8	8:8	1:64	4.747	1.029	65.855

7.5 Observations

Since each processing node is fully isolated from its neighbors, the program must explicitly set up the topology of the nodal layout. This implies that to change the size of the problem, or the size of the partition, the code must be recompiled.

Moreover, optimization did not seem to help (as per the previous results with the global use of CMF.) The small variations in time even indicate that no optimization is being accomplished. However, the substantially larger execution size of the code indicates that something indeed is being done - it is just that it does not appear to be beneficial.

The passing of boundary value information to fill in the ghost cells is by far the most time consuming. Since the topology is now arbitrary (from the compiler's point of view) it is impossible to do away with ghost cells; this is implicit in the use of message passing for a problem such as this. Certainly, global CMF was quickly (and implicitly) able to communicate between nodes using the *CSHIFT* command. Why is the CMF/CMMD inter-nodal communications so slow? Where is the time being spent in the message passing code?

TABLE 8. CMF/CMMD timings of gather/scatter operations, Z-Axis long (cmf$ layout)

B:NX	B:NY	B:NZ	Total Time	Gather	Message	Scatter	% Gather
5:8	5:8	1:40	50.63	39.00	2.34	6.11	77.03%
4:10	8:5	1:40	44.06	33.21	2.17	5.97	75.37%
8:5	4:10	1:40	44.00	33.29	2.01	5.98	75.66%
5:8	8:5	1:40	37.35	27.87	1.87	5.32	74.62%
8:5	5:8	1:40	37.29	27.86	1.82	5.32	74.71%
8:5	8:5	1:40	28.75	20.88	1.72	4.50	72.63%
4:16	8:8	1:64	134.18	106.23	4.98	15.07	79.17%
8:8	4:16	1:64	133.97	106.54	4.45	15.08	79.53%
8:8	8:8	1:64	77.00	59.10	3.49	9.92	76.75%

The answer is found with the separated gather/scatter runs. As mentioned earlier, within a message passing call there are three things that must be done: a gather of outbound data; a send/receive type message to the neighboring node; and a scattering of the inbound data. The timing studies of Table 8 and Table 9 show that most of the time is spent in the gather or message passing operation (typically 70-80%.) The break down given in Table 8 corresponds to the runs of Table 5 . The time is given in seconds, measured over the 24 time steps of the computer run. *Total Time* refers to the time spent within the message passing subroutines of the code. *Gather* refers to the time within these calls spent gathering data for the outbound messages. *Message* refers to the time actually spent in message communication with neighboring nodes. *Scatter* refers to the time spent scattering an inbound message to the internal data structure. *% Gather* is the percentage of time spent on the gather portion of the message passing.

Likewise, Table 9 corresponds to Table 4 . Here the time is mostly being spent in the message passing itself. The *% Message* column gives the percentage of time spent in this routine.

It appears that the *cmf$ layout* directive (when used in conjunction with the CMMD libraries) lays out arrays to make inter-nodal communications most efficient. Thus while the gather time is slow (indicating that the 2-D ghost cell array is not conformably placed on the same VP as the larger 3-D array) the message passing is fast. The opposite is true with the *cmf$ align* directive. In this case the arrays are conformal. The gather/scatter is fast since the two arrays are guaranteed to be on the same VP. But the message passing is slower, since the 2-D array is not aligned properly for most efficient message passing.

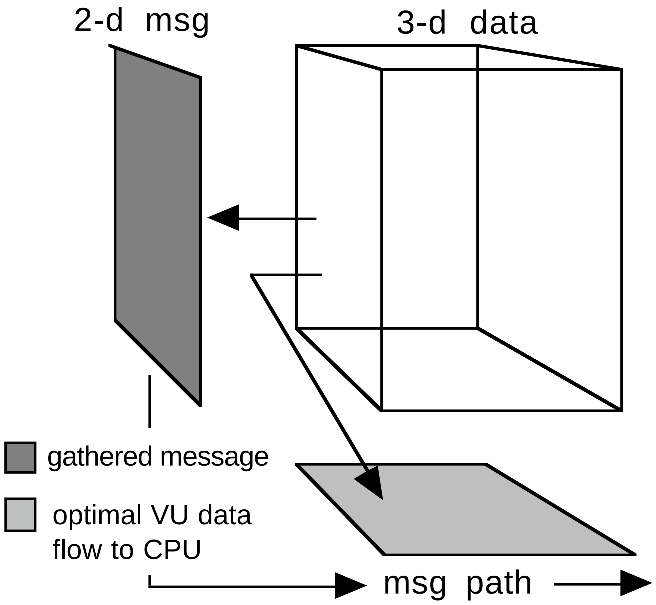
TABLE 9. CMF/CMMD timings of gather/scatter operations, Z-Axis long (cmf$ align)

B:NX	B:NY	B:NZ	Total Time	Gather	Message	Scatter	% Message
5:8	5:8	1:40	27.40	2.14	21.41	1.57	78.14%
4:10	8:5	1:40	23.34	1.81	18.30	1.30	78.41%
8:5	4:10	1:40	22.30	1.80	16.98	1.32	77.08%
5:8	8:5	1:40	19.56	1.52	15.31	1.09	78.27%
8:5	5:8	1:40	19.88	1.51	15.64	1.10	78.67%
8:5	8:5	1:40	15.12	1.12	12.01	0.81	79.43%
4:16	8:8	1:64	75.28	5.79	59.77	4.15	79.40%
8:8	4:16	1:64	70.99	5.83	55.44	4.16	78.10%
8:8	8:8	1:64	44.74	2.68	36.07	2.32	80.62%

An example is graphically presented in Figure 5. It is desired to slice off the left hand face for an outbound message. The 3-D data is spread out among the 4 VU. The gathered 2-D array (dark gray) is not oriented within the VU in the most optimal pattern to get it to the SPARC chip and the communication network quickest. This is the way the *cmf$ align* directive would place the data. Thus the gather is fast, but the message passing (due to the slowness of the VU getting the data to the SPARC etc.) is slow. On the other hand, the *cmf$ layout* directive appears to place the 2-D array in the optimal location (light gray); thus the gather is slow and the message passing is fast. Overall, for the case tested here, the *cmf$ align* directive improves code performance by a factor of approximately one and a half, and is thus the best choice.

Notice that in Table 8 if the gather times were comparable to the scatter times, the performance of the message passing version of this code would be substantially improved. For instance, the 8:8 by 8:8 by 1:64 run would increase from 1.39 to 3.33 MFlops/Node, corresponding to an overall rate increase from 88.86 to 212.80 MFlops. While it is still not comparable to the global CMF rates, it is a substantial increase. The same comment holds with the data of Table 9 ; bringing down message passing times to more optimal levels of Table 8 would substantially increase overall performance.

FIGURE 5. Data Flow Example



8.0 Fortran 77 with Message Passing

The final approach examined here uses each node as a SPARC chip. Fortran 77 (or C, etc.) is used to write a program in a similar manner to the parallel Fortran on a node approach discussed above. CMMD is used to establish communication of boundary values between nodes. This method is evaluated here to see if the VU as used in the message passing environment actually prove to be a major benefit.

This method is of interest in that it is the most portable. In fact, since it is written in Fortran 77 (for which a compiler exists for most machines) only the message passing libraries need be ported to a new machine. Many communities have codes that use this method for just this reason.

All the comments about domain decomposition and ghost cells still applies to this programming model. The only difference is that Fortran 77 is used to program each node instead of CMF.

8.1 Execution time results

Timing runs were made on partitions of various sizes including 32, 64, and 128. The grid sizes used in the computational problem included 40^3 and 64^3.

The same type of runs which were performed for the CMF/CMMD data sets were performed for the F77 runs. This will allow a direct comparison between the F77/CMMD and CMF/CMMF versions of the codes.

TABLE 10. F77/CMMD with Z-Axis long and communication timings

B:NX	B:NY	B:NZ	Time/Step	% Comm.	Mflops/ Node	Total Mflops	Comm. Time/word
5:8	5:8	1:40	2.174	17.87%	1.406	35.150	1.493
4:10	8:5	1:40	1.740	20.22%	1.380	44.151	1.713
8:5	4:10	1:40	1.737	20.13%	1.382	44.230	1.358
5:8	8:5	1:40	1.374	22.64%	1.402	56.081	1.366
8:5	5:8	1:40	1.373	22.04%	1.403	56.127	1.338
8:5	8:5	1:40	0.908	25.63%	1.337	85.597	1.303
4:16	4:16	1:64	13.776	12.07%	1.400	22.402	1.897
4:16	8:8	1:64	6.916	14.97%	1.403	44.900	1.662
8:8	4:16	1:64	6.897	14.92%	1.407	45.023	1.627
8:8	8:8	1:64	3.480	19.71%	1.404	89.834	1.407

Table 10 includes communication timing results. Other than this distinction, the nodal configurations are identical to the runs examined in Table 4 or Table 5 . Remember that with F77 there are no directives for the layout of arrays (i.e, there is no *cmf$ layout* or *cmf$ align* directives in F77.)

TABLE 11. F77/CMMD with X-Axis long

B:NX	B:NY	B:NZ	Time/Step	Mflops/ Node	Total Mflops
1:40	5:8	5:8	2.396	1.276	31.890
1:40	4:10	8:5	1.812	1.324	42.382
1:40	8:5	4:10	1.815	1.323	42.326
1:40	5:8	8:5	1.466	1.314	52.560
1:40	8:5	5:8	1.494	1.290	51.584
1:40	8:5	8:5	1.081	1.124	71.906
1:64	4:16	4:16	13.851	1.393	22.280
1:64	4:16	8:8	6.907	1.405	44.956
1:64	8:8	4:16	6.831	1.420	45.454
1:64	8:8	8:8	3.497	1.397	89.404

Table 11 and Table 12 illustrate the difference that the geometry of layout makes upon the Fortran 77 runs. The layout-geometries of the runs are identical to those reported in Table 6 and Table 7 .

TABLE 12. F77/CMMD with Z-Axis long

B:NX	B:NY	B:NZ	Time/Step	Mflops/ Node	Total Mflops
5:8	5:8	1:40	2.236	1.367	34.177
4:10	8:5	1:40	1.692	1.419	45.409
8:5	4:10	1:40	1.683	1.426	45.644
5:8	8:5	1:40	1.424	1.353	54.102
8:5	5:8	1:40	1.487	1.295	51.818
8:5	8:5	1:40	0.873	1.392	89.083
4:16	4:16	1:64	13.560	1.422	22.759
4:16	8:8	1:64	6.898	1.407	45.017
8:8	4:16	1:64	6.898	1.407	45.014
8:8	8:8	1:64	3.414	1.431	91.574

8.2 Observations

It is important to note that these F77/CMMD runs give performance similar to the CMF/CMMD runs. The CMF/CMMD runs have the benefit of using the VU, while the F77/CMMD runs do not. This difference is clearly a result of the excessive amount of time the

CMF/CMMD runs take to perform the gather or the message passing operation. To see this clearly, compare the *% Comm.* column of Table 10 to those of Table 4 and Table 5 .

The break-down of these separate gather, message passing, and scatter operations were also performed for the F77/CMMD runs (see Table 13 .) When compared to Table 4 and Table 5 it quickly becomes clear that F77 is much better able to handle the gather type operations and message passing than is CMF.

In these runs, the times for the gather operations and the scatter operations are comparable. Likewise, the message passing is not an order of magnitude more expensive than the gather/scatter. This represents quite a savings in time resulting in the fact that these runs are overall basically as fast as the CMF/CMMD runs.

TABLE 13. F77/CMMD timings of gather/scatter operations, Z-Axis long

B:NX	B:NY	B:NZ	Total Time	Gather	Message	Scatter	% Gather
5:8	5:8	1:40	9.43	2.68	4.51	2.45	28.42%
4:10	8:5	1:40	8.49	2.47	3.37	2.26	29.09%
8:5	4:10	1:40	8.41	2.53	3.20	2.30	30.08%
5:8	8:5	1:40	7.35	2.14	2.93	2.03	29.12%
8:5	5:8	1:40	7.47	2.18	3.00	2.04	29.18%
8:5	8:5	1:40	5.89	1.71	2.39	1.59	29.03%
4:16	4:16	1:64	39.24	10.70	18.42	9.14	27.27%
4:16	8:8	1:64	25.13	6.91	11.57	6.08	27.50%
8:8	4:16	1:64	24.80	6.98	11.05	6.18	28.15%
8:8	8:8	1:64	15.42	4.27	6.80	3.99	27.69%

9.0 Timing Results Conclusions

All other aspects being equal, the global CMF is by far the better choice. It is faster than both CMF/CMMD and F77/CMMD by an order of magnitude in raw speed. If message passing is desired for other reasons (such as having different portions of code executing on various nodes) then there is not a clear winner between F77/CMMD and CMF/CMMD. If, as expected, the bottleneck of the excessively long gather or message passing times is cleared up, then CMF/CMMD would be the winner due to its ability to take advantage of the fast floating operations inherent in the VU. However, for now, the portability advantages of Fortran 77 as opposed to CMF make F77/CMMD the marginal short-term winner among the message passing codes.

Simulation of the Rayleigh-Taylor Instability

10.0 Introduction

The Rayleigh-Taylor instability is the name given to one of the classic problems of fluid mechanics; a heavy fluid situated above a lighter fluid in a gravitational field. With gravity directed downwards, the heavy fluid will try to move downwards (falling) and the lighter fluid will try to go upwards (buoyancy.) This problem is of interest to many of the communities that study fluid mechanics. In the geophysical community, this situation is encountered in both the atmosphere and ocean. Here the different densities can be caused by gradients of temperature or salinity (in the case of the ocean.) The combustion and reactive flow communities are interested in the Rayleigh-Taylor instability; non-premixed systems will have regions of contact between the fuel and oxidizer where sharp density gradients can occur, resulting in Rayleigh-Taylor like mixing if the more dense component is on top. Turbulent mixing is also often the case here if the relative velocities are large enough. The laser community is also interested in the Rayleigh-Taylor instability, and the similar Richtmyer-Meshkov instability[1]. Instabilities of this sort arise with the impact of a laser beam on a surface of a fuel pellet, a problem of interest to those in the laser community who study inertial confinement fusion. Finally, the astrophysical community studies unstable stratified convective layers which are relevant to the behavior of stellar interiors. In this case, heavy layers of stellar matter overlay less dense layers. The strong gravity and electro-magnetic field of the star can drive substantial instabilities in this case. While not exactly the same as the purely gas- or hydrodynamic case of this dissertation, the study of the Rayleigh-Taylor instability (and its hydromagnetic equivalent) is often used to shed light on this astrophysical problem.

Most of the early literature on the Rayleigh-Taylor instability is for incompressible fluids. The early experiments were done with liquids which are effectively incompressible. Much of the more recent work is for compressible fluids. This is certainly of more interest to the combustion and laser communities, which deal with more extreme cases of density and

[1] Here there is no gravity; a shock wave (impulse) initiates the mixing effects.

temperature gradients. The classic reference for all instability problems is Chandrasekhar [20]. A more recent general reference on instabilities is Drazin and Reid [27]. Both of these tend to focus on incompressible instability problems however. The original work and experiments on stratified density instabilities are reported in Rayleigh [85][86] and Taylor [100] (from which the Rayleigh-Taylor instability gets its name.) while more recent experimental results are in Read [87]. A general overview of the Rayleigh-Taylor instability is presented in Sharp [95], which contains a review of the work done through the mid 1980's. The basic references for the compressible Rayleigh-Taylor problem are the works of Book and Bernstein [16][17][18]. The "modern" Rayleigh-Taylor experiments were performed by Read and Youngs, who used a rocket motor apparatus to produce a strong gravity field. The experimental results are reported in Read [87], and a computational model developed in parallel with these experiments is reported in the works of Youngs [112][113][114]. The laser and inertial confinement fusion communities have also contributed greatly to the experimental study of the Rayleigh-Taylor instability. This research is representatively documented in Remington, Glendinning et. al [88][89][38][90].

The rest of this chapter assumes a familiarity with the Rayleigh-Taylor problem, and the linear analysis of the compressible equations. "Appendix D - Linear Rayleigh-Taylor Theory" contains an extensive review of the assumptions, derivations, and final results of the linear perturbation theory for a sharp interface. All variables used conform with those listed in Table 16, "Common Variables," on page 84.

10.1 Purpose of this research

Using the data parallel Fortran code developed and discussed in the previous chapter, a study will be made of the Rayleigh-Taylor instability for compressible fluids. The three-dimensional Navier Stokes equations of Section 17.4 on page 86 will be solved using the two-step MacCormack method of Section 24.2 on page 114. With this computational code, it is desired to make a parameter study of the random mixing layer of a density stratified fluid. Basically, this is a fluid or gas where a higher density fluid is placed above a lighter fluid, and small random fluctuations initiate mixing of the two substances due to the inherently unstable interface.

In order to study this problem, several steps will need to be taken. These steps, which constitute the bulk of this chapter, include:

1. Develop a scheme to assist the MacCormack method in dealing with the density discontinuity of the Rayleigh-Taylor problem.
2. Test the resultant method against known solutions to verify its correctness.
3. Develop a scheme to initiate the random fluctuations in a natural, non-intrusive way.

The first step is required since the Lax-Wendroff finite difference methods (of which the MacCormack scheme is an example) handle discontinuities poorly. This is more fully detailed in Section 24.3 on page 116. The method of solution presented in this dissertation is of two parts. The first part is the inclusion of a Smagorinsky eddy-viscosity in the Navier-Stokes equations. This acts as an artificial viscosity, which helps to dampen the unwanted “ringing.” More details of the Smagorinsky eddy-viscosity can be found in Section 22.1 on page 101, while the general effects of artificial viscosities can be seen in a brief example in Section 24.3 on page 116. The second part of the solution is to develop a technique to “smooth out” the interface in a mathematically correct manner. This is detailed in Section 11.0 on page 33.

It is next necessary to check the veracity of the resultant computational code. To do this, the code will be tested against the known analytic solution of the linear perturbation problem (see “Appendix D - Linear Rayleigh-Taylor Theory” for the details of this solution.) This is done in order to develop a good understanding of the strengths and weaknesses of the effects of the eddy-viscosity terms, and the smoothing. To this end, a number of trial computations are presented. These will show that the MacCormack method can handle the Rayleigh-Taylor problem within certain restrictions of density discontinuities and the amount of smoothing to this jump that is done. However, it will still not fully overcome a known weakness of the MacCormack scheme; dispersive errors still generate errors in the calculations of velocities. These tests and results are presented in Section 12.0 on page 43.

The final step is the development of a scheme to initiate the random mixing. The method used here is the natural outcome of an extension to the eddy-viscosity model mentioned above. One of the weakness of a scheme such as the Smagorinsky eddy-viscosity is that while it removes energy from the resolved computational length scales (i.e., everything that can be computed with the grid size chosen,) there is no mechanism to put energy from these unresolved scales back into the system. Recent research has come up which a scheme to do this: stochastic backscatter. Stochastic backscatter, as used in this dissertation to improve the Smagorinsky eddy-viscosity, is described in Section 22.2 on page 106. However, besides being an addition to the eddy-viscosity, the random forcings of the stochastic backscatter can be made to serve another purpose; these random accelerations will be used to break the symmetry of the initial hydrostatic unstable state, and initiate random mixing. Once this is done, it is possible to check the resultant mixing against known scaling and growth properties, which have been observed in both physical and computational experiments. This is done in Section 14.0 on page 62, where the agreement is very good.

10.1.1 Previous research on this problem

The computational work in this dissertation is motivated by a distinguished series of previous research. The two-dimensional work in Gardner et. al [36] studied the linear perturbation problem using both statistical models and a finite difference method including a front tracking scheme to explicitly resolve the interface between the two fluids. This work was extended to three-dimensions in Tryggvason and Unverdi [103]. The numerical scheme used in these papers is more complicated than the MacCormack scheme used here. The advantage of this simpler scheme is that it is easily parallelilized, thus allowing it to take advantage of the new massively parallel computers. The trade-off is that the more complicated schemes resolve the velocity field better.

Li [65] works with a different three-dimensional scheme: the level set (or function) method added to a TVD scheme. Li’s method, like the method of this dissertation, uses a massively parallel computer. However, he uses domain decomposition techniques on an Intel/IPSC-860 hypercube. He gets very good results for the linear perturbation problem in three-dimensions. However, his work on the randomly perturbed interface is only preliminary, as compared to the detailed analysis presented in this dissertation.

For the three-dimensional study of the randomly perturbed interface, that of Youngs [114] is the most thorough. The code used in this study is run vectorized and in parallel on a Cray-YMP. However, Youngs does not use an explicit “subgrid-scale” model to handle the dissipation. He allows the numerics itself to provide any dissipative mechanisms needed to keep the computation reasonable. While this worked very well for his computations, the author of this dissertation is much more comfortable with schemes where the details are explicitly spelled out instead of leaving it to the numerics to “do the right thing.” This dissertation uses the large-eddy simulation concept to handle all effects at a scale larger than the grid size, and a subgrid-scale model to handle every thing smaller. And furthermore, this subgrid-scale model includes stochastic backscatter. This provides the mechanism for adding random fluctuations to get the mixing started, without having to specify any further perturbations.

10.2 Outline of this chapter

The rest of this chapter contains the results of the research outline in Section 10.1 on page 30. The next section contains the derivation of the equations used to describe a smoothly varying density transition in an unstably stratified fluid. After this, the results of the test calculations for the linear perturbation problem are presented. The final section contains the ultimate goal of this chapter: the results from the random mixing layer problem using stochastic backscatter to initiate the mixing.

11.0 The Smooth Hydrostatic Interface

One of the weaknesses of the Lax-Wendroff method is its inability to handle sharp interfaces. This includes shocks arising from supersonic flow and density discontinuities. The density and internal energy interfaces in the Rayleigh-Taylor problem are of course just such a discontinuity. At discontinuities of this nature, Lax-Wendroff methods generate severe ringing (Gibbs-like phenomena) as a result of numerical truncation errors. Lax-Wendroff methods are very dispersive (see “Appendix C - Finite Difference Methods” for a more complete discussion of the causes of the non-physical propagation of acoustic waves away from sharp interfaces.) There are several methods to alleviate this problem:

- use a “better” numerical method (FCT, TVD, front tracking, etc.)
- smooth out the sharp interface

- add an artificial viscosity term

This dissertation explores the latter two approaches, in an attempt to see how much can be accomplished with them.

The standard linear perturbation analysis for the Rayleigh-Taylor problem assumes a sharp interface.[2] It is also possible to construct an interface that is "smoothed out," in which the density and internal energy profiles no longer have a sharp discontinuity at the "interface." While it is easy to generate the hydrostatic solution for this case, the corresponding perturbational linear solution is less obvious. One method possible is to construct a series of small jumps that approximate the smooth interface. Book [18] has worked this out in detail. The final perturbation is a superposition of the linear parts arising from each small step. While this method could be used in a computational scheme, the actual analytic work is very substantial due to the non-linear solve necessary to determine the growth rate for each little step.

The following is the solution of the hydrostatic problem for a smooth interface (see Section 25.1 on page 122 for a derivation of the hydrostatic problem with a sharp density jump.) The smoothed interface will take on a hyperbolic tangent nature, which in a limiting process is still a sharp interface.[3]

It is desired that the internal energy have the properties of Figure 48, namely that at the lower end the internal energy is asymptotic to $\frac{\hat{u}_0}{S}$, and at the upper end to $\hat{u}_0$. S is the density jump at the interface (defined as the ratio of large to small densities.) Since the function is to be smooth, the value exactly at the interface $z = z_i$ will be the average value of these two. One function that can satisfy these requirements is the hyperbolic tangent function. Thus the definition of internal energy is:

$$\hat{u}(z) = \hat{u}_0 \left\{ \frac{1}{S} + \left[\frac{1}{2}\left(1 - \frac{1}{S}\right)\right] [\tanh(\alpha(z - z_i)) + 1] \right\} \qquad \text{(EQ 4)}$$

[2] See "Appendix D - Linear Rayleigh-Taylor Theory" for a complete derivation.

[3] In the limit this goes to a step function as before.

where α is the "smoothing parameter."[4] In the limit $\alpha \to \infty$, this goes to the step function of Figure 48 and Equation 90. In the limit $\alpha \to 0$ it becomes almost a straight line connecting $\frac{\hat{u}_0}{S}$ and $\hat{u}_0$. Acceptable values of α are in the range 15 to 50, where the interface is still reasonably sharp, but the Lax-Wendroff schemes no longer see a discontinuity (this will be computationally demonstrated in the next sections.)

When this value of $\hat{u}(z)$ is used to solve the hydrostatic balance (as is done in Section 25.1 on page 122 for the sharp interface,) the resultant pressure equation is:

$$P(z) = P_0 \exp\left(\frac{g(z-z_i)}{(\gamma-1)\hat{u}_0}\right) \exp\left(\frac{g(1-S)}{2(\gamma-1)\hat{u}_0\alpha} \ln\left[1 + \frac{1}{S}\exp(-2\alpha(z-z_i))\right]\right) \qquad \text{(EQ 5)}$$

Reference pressure P_0 must be scaled correctly to get density ρ_0 on one side of the jump, and $\frac{\rho_0}{S}$ on the other. To do this, the condition

$$P(z_i) = \frac{(\gamma-1)\rho_0\hat{u}_0}{S} \qquad \text{(EQ 6)}$$

must be met. By substituting $z = z_i$ into Equation 5, we find that

$$P_0 = \frac{(\gamma-1)\rho_0\hat{u}_0}{S} \exp\left(\frac{-g(1-S)}{2(\gamma-1)\hat{u}_0\alpha} \ln\left[1 + \frac{1}{S}\right]\right) \qquad \text{(EQ 7)}$$

Finally, the corresponding density equation is recovered from the equation of state:

$$\rho(z) = \frac{P(z)}{(\gamma-1)\hat{u}(z)} \qquad \text{(EQ 8)}$$

where $P(z)$ is defined in Equation 5, and $\hat{u}(z)$ is defined in Equation 4.

11.1 Test Computations

The following graphs show a comparison of the amount of smoothing caused by three separate values of α. The three-dimensional Navier Stokes equations of Section 17.4 on page 86 are integrated using the two-step MacCormack method of Section 24.2 on page 114 in a non-dimensional form with g=.25, S=2, and at a CFL condition of 0.8. The

[4] Not to be confused with the constant $\alpha_\pm$ from the linear perturbation theory analysis!

initial conditions are as defined in the previous section. The physical vertical domain size is 8, with a discretization of 256 internal grid points. For each run, the initial condition at t=0 and the final condition at t=4 are shown. The fluid is an ideal perfect gas with insulated free-slip lids on top and bottom, i.e.,

$$\left.\frac{\partial u}{\partial x}\right|_{\mathrm{lid}} = 0 \qquad \left.\frac{\partial v}{\partial y}\right|_{\mathrm{lid}} = 0 \qquad \left.w\right|_{\mathrm{lid}} = 0 \tag{EQ 9}$$

and the fluid has infinite extent in the horizontal plane. The interface between has no perturbations; hence the state of the gas while unstable should be stationary. The first interface has $\alpha = 1$; this is severely smoothed out with very little evidence of a density jump at all. The second run has $\alpha = 25$; the density jump is very evident, but the smoothing still occurs over a significant number of grid points. The final run has $\alpha = 1000$; here the interface is almost a perfect step function.

Several different interesting features are evident from these computations. In all three cases by just viewing density, pressure, and internal energy one would conclude at the gross level that the fluid was in a stationary state (as it should be for this hydrostatic stationary state - there are no perturbations to start fluid motion.) However, by looking at the vertical velocity component w, it is evident in all three cases the fluid is far from stationary. What is the cause of this non-physical motion?

The cause is quite simple, but subtle. Since the interface is perfectly flat, it is not any built-in perturbation that is causing the motion. It is the discrepancies between the numerical and analytical forms of the equations and values that are causing the perturbation! Due to round off errors in representing the values of the functions, and truncation errors in representing the equations, it is impossible to represent the continuous analytic hydrostatic balance exactly with a discrete computer code of finite-order accuracy. The Navier-Stokes equations and boundary values can not be represented exactly with a finite difference method (the MacCormack method used here is only second-order accurate.) These calculations were done with IEEE double precision arithmetic; only a finite number of discrete rational values can be represented, as opposed the continuous range of values any variable such as density could actually take.

FIGURE 6. $\alpha = 1$ **Hydrostatic Initial Condition**

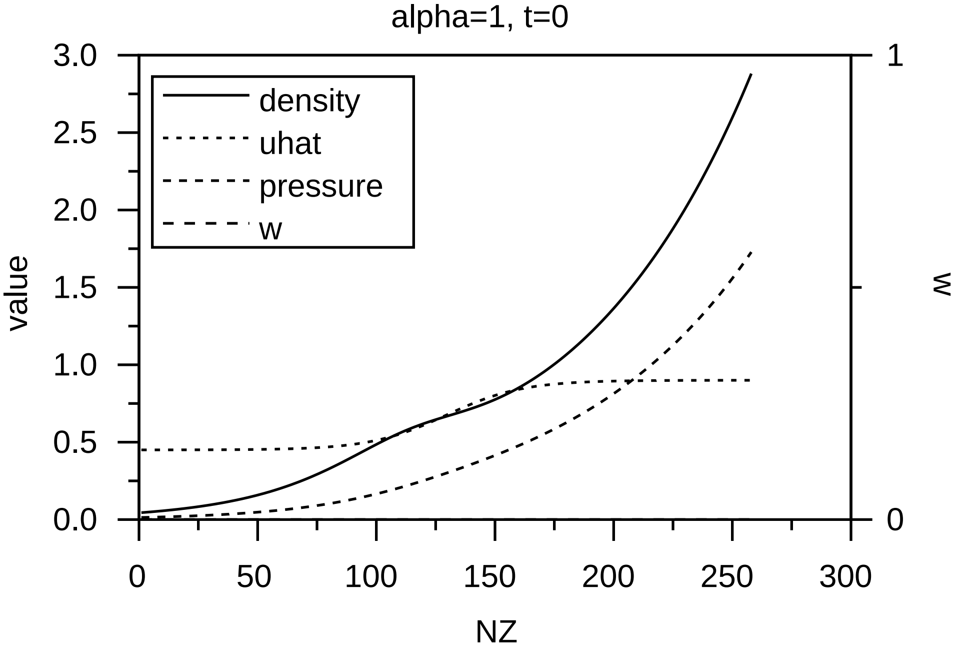


FIGURE 7. $\alpha = 1$ **Hydrostatic Final Condition**

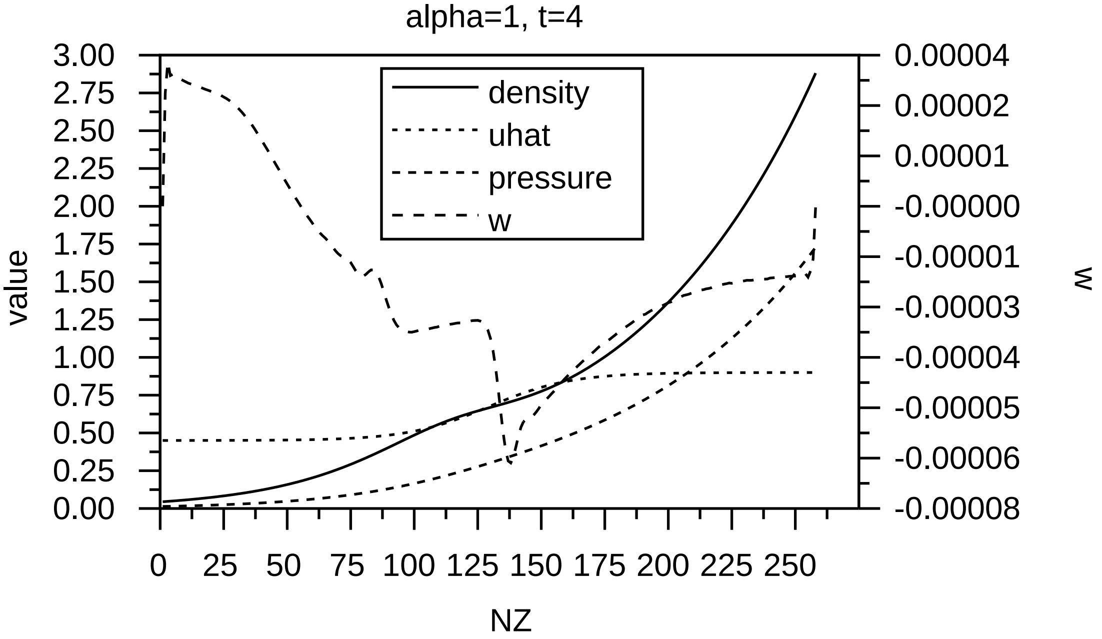


FIGURE 8. $\alpha = 25$ **Hydrostatic Initial Condition**

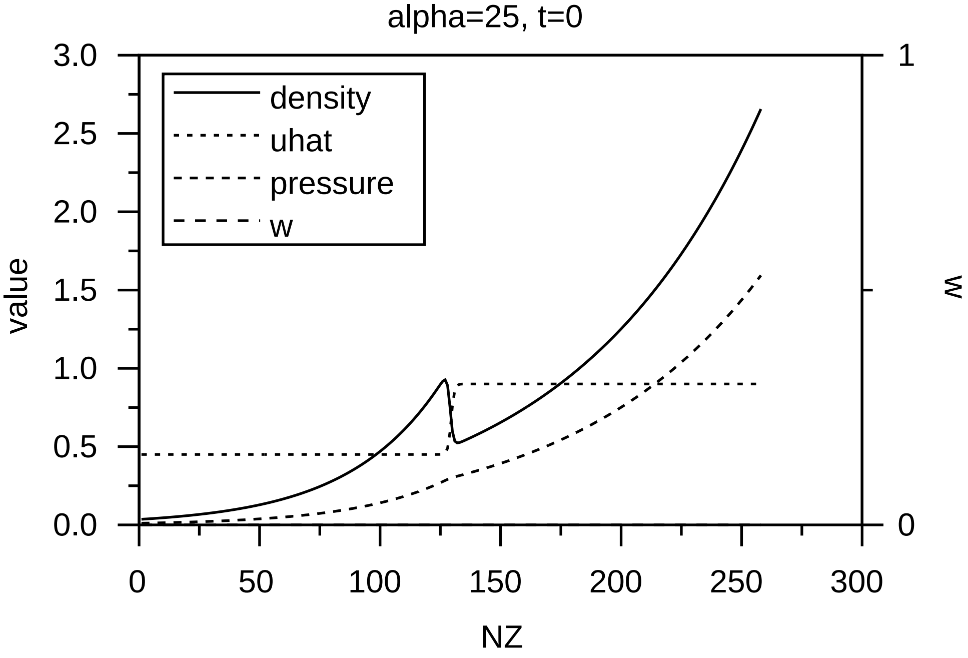


FIGURE 9. $\alpha = 25$ **Hydrostatic Final Condition**

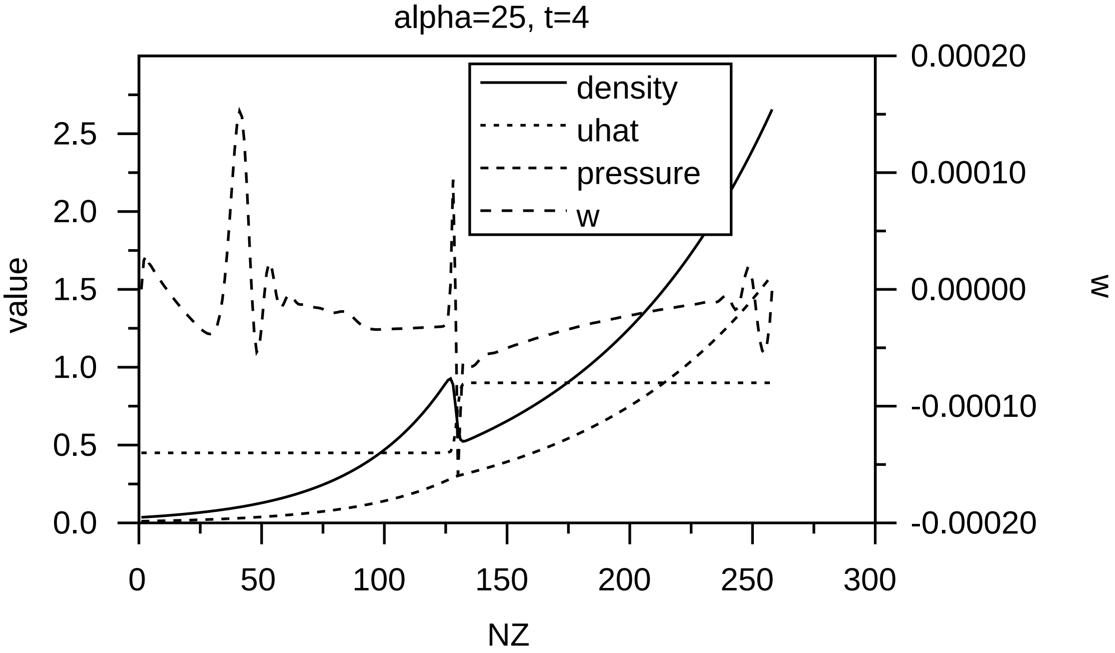


FIGURE 10. $\alpha = 1000$ **Hydrostatic Initial Condition**

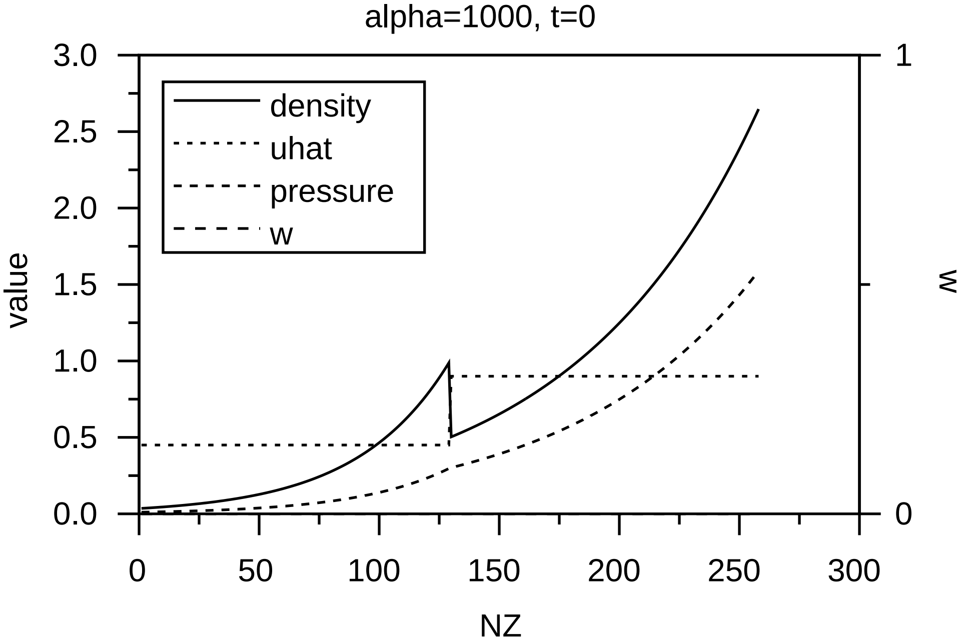
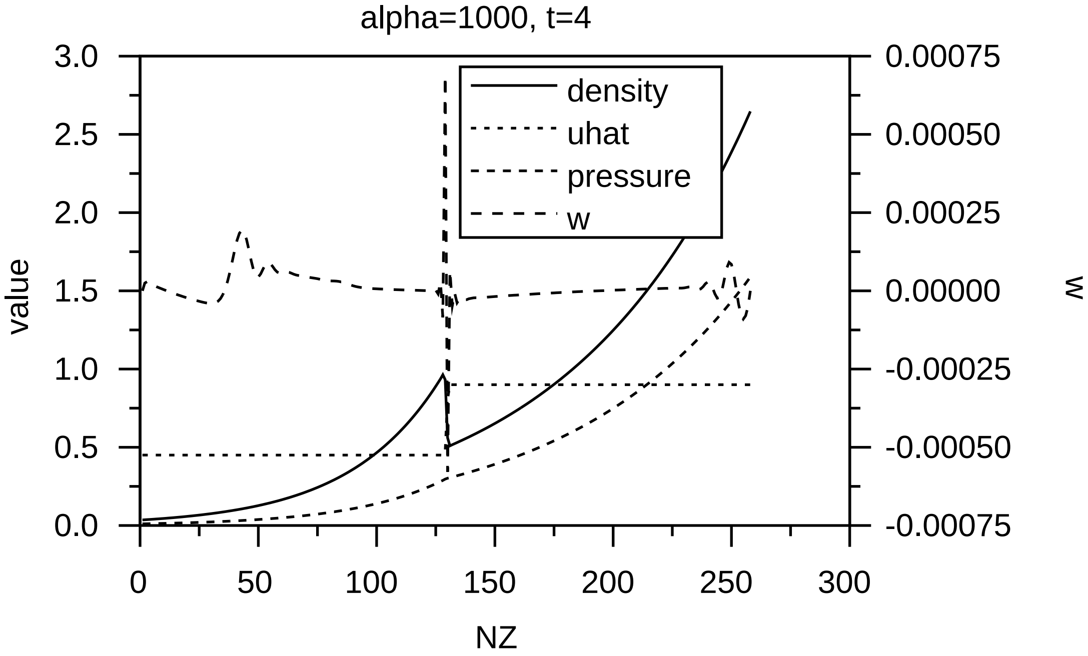


FIGURE 11. $\alpha = 1000$ **Hydrostatic Final Condition**



The net result of these *discretization errors* is that there is a net positive flow (towards the interface) above the interface, and a net negative flow (also towards the interface) below the interface. Fortunately the magnitude is quite small (especially when compared to the velocities generated by perturbations;) Figure 7 clearly illustrates the net flow, and its relative magnitude. There is also a pulse generated at the boundaries due to the same type of error process. These pulses travel as acoustic waves (i.e., at the local sound speed) towards the interface. The magnitude is larger but of the same order of magnitude as the overall net positive/negative flow.

There is another error in the velocity field. This is the error generated at the interface by the numerical scheme. Lax-Wendroff methods have substantial difficulties handling interfaces. Dispersive errors cause a series of acoustic pulses to be sent out from the sharp interface.[5] Notice that as the interface is smoothed out, these pulses become much smaller. When the interface is not smoothed out, the size of this pulse is orders of magnitude larger than the errors of the previous paragraph (Figure 11.) But as the interface become more smooth, this error is on the same order (Figure 9,) then indistinguishable from the other errors (Figure 7.) This is, of course, the reason to smooth out the interface in the first place. It is desired to keep the interface as sharp as possible, while smoothing it out enough so that these dispersive errors are no worse than the other errors inherent in the calculation. By the above calculations, it would appear that $\alpha = 25$ is the right compromise value.

The above calculations done on a stationary hydrostatic state give an indication of the best results achievable with any calculation of the Rayleigh-Taylor instability. The linear theory assumes only one perturbation (with a sharp interface) is put into the calculation. The above calculation shows that there are more (uncontrollable) perturbations put in by these errors, and that the interface can not be too sharp. When analyzing the results of the next sections it is important to remember these sources of error, and the fact that the base state itself is not truly numerically stationary.

[5] More details of these dispersive errors are given in "The Lax-Wendroff Method" on page 112.

11.2 Effect of a smooth interface on the linear perturbation theory

The original analytic equations derived from the linear theory assumed a sharp interface. But the previous section illustrated that a sharp interface could not be achieved with a code based on the Lax-Wendroff methods. A few more words on this are in order.

In theory, it is possible to take the stationary solution (Equation 4, Equation 5 and Equation 7) and repeat the analysis of Section 25.1 on page 122. If an analytic solution were possible, this would give an exact new set of perturbation equations (equivalent in form to Equation 99- Equation 102.) However this does not seem likely. The highly non-linear form of Equation 4, Equation 5 and Equation 7 does not lend itself to achieving a reasonable Sturm-Liouville problem which can be analytically solved by a separation of variables. Even if this were possible, the resultant non-linear root-solve needed to compute the growth rate (the new analog to Equation 104) would be extremely difficult, if not impossible, to solve. Thus even if the Sturm-Liouville problem could be solved numerically, the problem of finding the growth rate still remains.

Book [18] gives a mathematical result for an interface made up of a series of step functions. In practice, the smooth interface could be cast into this format. However, with the interface being spread out over approximately ten grid points, this would necessitate a numerical calculation. This would be extremely tedious, and the difficulty of determining the net growth rate still remains. In total, this is an area that still needs further examination.

Thus a smoothed out interface added to the perturbations of the linear theory (derived with a sharp interface) represents a source of error. How serious is this error?

This is in fact somewhat difficult to determine. Given that the numerical method inserts an error at the same time as does a smoothed out interface, it becomes impossible to separate the two errors. Add to this the fact that the Lax-Wendroff methods have dispersive truncation errors, resulting in yet more variance between analytic and observed growth rates! The only analysis possible are test calculations such as those done in the previous section. These give a feel for the order of magnitude of the errors from each part. For the very sharp interface, the entire errors will be the results of the deficiencies of the numerics (since this matches the linear theory.) For the smoothed out interface, any discrepancies will be less due to the numerics, and more due to the difference between a growth rate derived from a sharp interface being used in the linear analysis whereas the interface is actually smooth.

The final fact to take into account is the thickness of the interface as compared to the wavelength of the perturbation. The wavelength of the perturbation is the dominant length scale that enters into the growth rate calculation. A “back of the envelope” calculation is in order. For a smoothing parameter of $\alpha = 50$, the interface is spread out over approximately ten zones, or a non-dimensional length of 0.3125. The wavelength in the above calculations is one. This represents a transition region of on the order of 30% of the wavelength, which corresponds to a variation in wavenumber of 4.83 to 8.98. In Figure 51, “Growth Rate as a Function of Wavenumber,” on page 130, for g=1.0, this corresponds to a growth rate in the range of 0.45 to 0.6. But for a smaller gravity of 0.125 the spread is from 1.0 to about 1.5. For large values of gravity, shocks form which the Lax-Wendroff methods can not handle[6], so the calculations are limited to the lower end of the band. This is of course worse for the errors in growth rate!

In summary, there will be four interwoven sources of errors in the following calculations done with the two-step MacCormack method:

- Overall discretization errors of the initial conditions and equations
- Errors introduced by discretization errors at the boundaries
- Dispersive errors at any sharp interfaces/discontinuities
- Errors in the actual growth rate caused by the smoothing of the interface

[6] There are computational examples in Section 12.2 on page 46.

12.0 The Linear Theory Calculations

The following calculations are done to help validate the code running on the CM-5, and to check its performance against other current codes such as Li [65], and Youngs [114]. Several different results will be reported. The effect of the smoothing parameter on a 2-D perturbation will be explored, and compared against the analytic result for a sharp interface. The effect of gravity will be explored, also with a 2-D perturbation. Sample calculations will compare a 2-D perturbation against a 3-D perturbation. Next calculations are done to observe known long time behavior of 3-D perturbations. Finally, the differences induced by the eddy-viscosity model will be examined.

All the following calculations are done in dimensionless units. The three-dimensional Navier Stokes equations of Section 17.4 on page 86 are integrated using the two-step MacCormack method of Section 24.2 on page 114. The initial conditions are the smoothed interface hydrostatic state of Section 11.0 on page 33 with the addition of linear perturbations of Section 25.2 on page 125. The base scale gravity and densities are unity ($g_z = 1$, $\rho_0 = 1$,) as is the base velocity scale which is set by the hot gas sound speed ($c_H^2 = 1$.) The computational domain is periodic in the horizontal x- and y-axes, with rigid, fixed, insulating, free-slip lids on the vertical z-axis. The length scale is set by the non-dimensional size of the domain ($1 \times 1 \times 8$,) with a computational grid of $32 \times 32 \times 256$. In all cases, the wavenumber of the perturbation is $\kappa = 2\pi$ (a wavelength of $\lambda = 1$,) and the amplitude is $A_x = 0.001$. The Atwood number is $A = \frac{1}{3}$, which corresponds to a halving of density at the interface ($S = 2$.) The interface is in the center of the vertical domain, with gravity pointing in the direction of the z-axis. All calculations are done with double-precision arithmetic, at a CFL condition ($\frac{c_H \Delta t}{\Delta x}$) of between 0.4 and 0.8[7]. Except where explicitly noted, there is no artificial viscosity or eddy-viscosity used in the calculations.

In calculations where the position and/or velocity of the interface is presented, the said value is determined by third-order interpolation. Since the MacCormack method does not explicitly track the interface, a passive scalar marker[8] is introduced whose zero value is

[7] The CFL condition used was selected to ensure stability of the calculation.

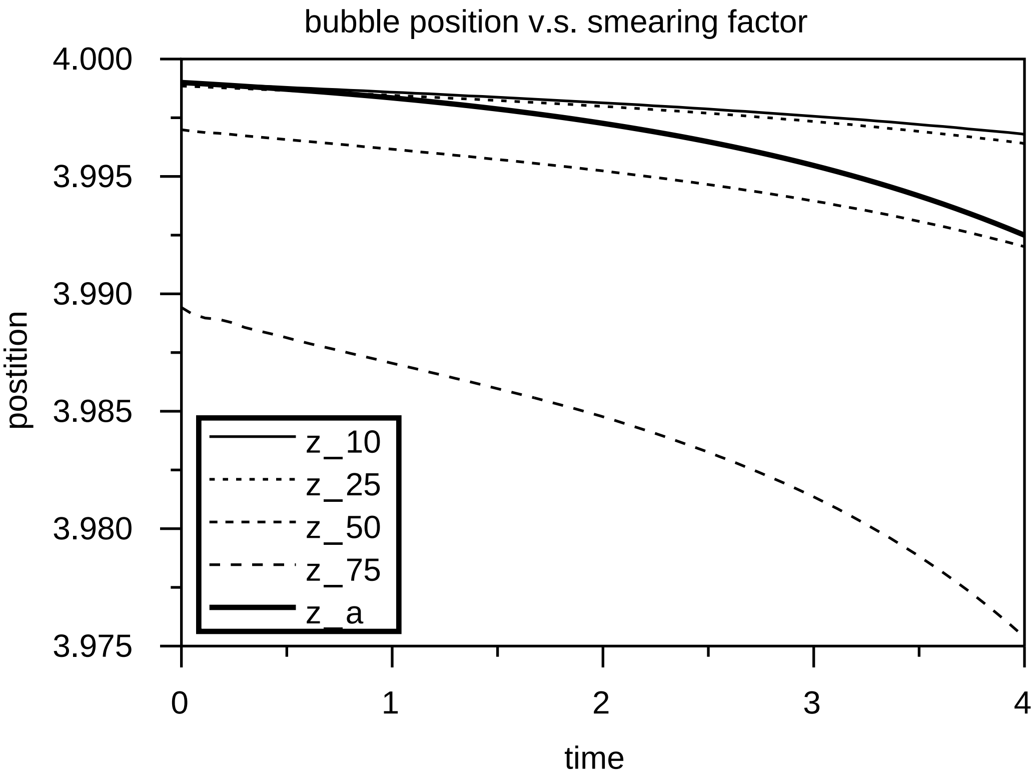
[8] See “Passive Scalars” on page 96.

placed at the interface $z = z_i$. This passive scalar is +1 at one end of the vertical domain, and -1 at the other, and smoothly (via the smoothing parameter) passes through zero at the interface. As the velocity field advects this passive scalar, the zero crossing point remains located at the interface (within round-off errors and accuracy of the advecting velocity field.) While the passive scalar is simply a diagnostic tool and has no physical significance, it can be thought of as a color marker added to the two fluids. One fluid is colored red for instance, and the other blue. By “watching” the color intermix, it is possible to observe the behavior of the flow, most specifically the effects of the velocity field on the colored fluids. All data are sampled at the center of the light fluid bubble, which is centered in the middle of the horizontal computational domain(i.e., the vertical center-line.) Remember that gravity points in the same direction as the z-axis (if the z-axis points upwards, the bubbles go down.)

12.1 The Effect of the Smoothing Parameter

This first set of data shows the effect of the smoothing parameter upon the position of the interface. Figure 12 shows the effect of various values of the smoothing parameter α on the position of the bubble over time. Values of α from 10 to 75 are tested. The heavy line represents the analytic solution. Note that errors in the interpolation algorithm give a slight variation in the position of the bubble; this is why all of the bubbles do not start at exactly the same correct starting value of 3.999, but have a variation of up to .009 (about .23%.) Without any artificial viscosity terms, the MacCormack method is not able to handle interface much sharper than for $\alpha = 75$, in fact at around 100 (which corresponds to a spread of about 3 cells) the acoustic pulse quickly drove the calculation unstable.

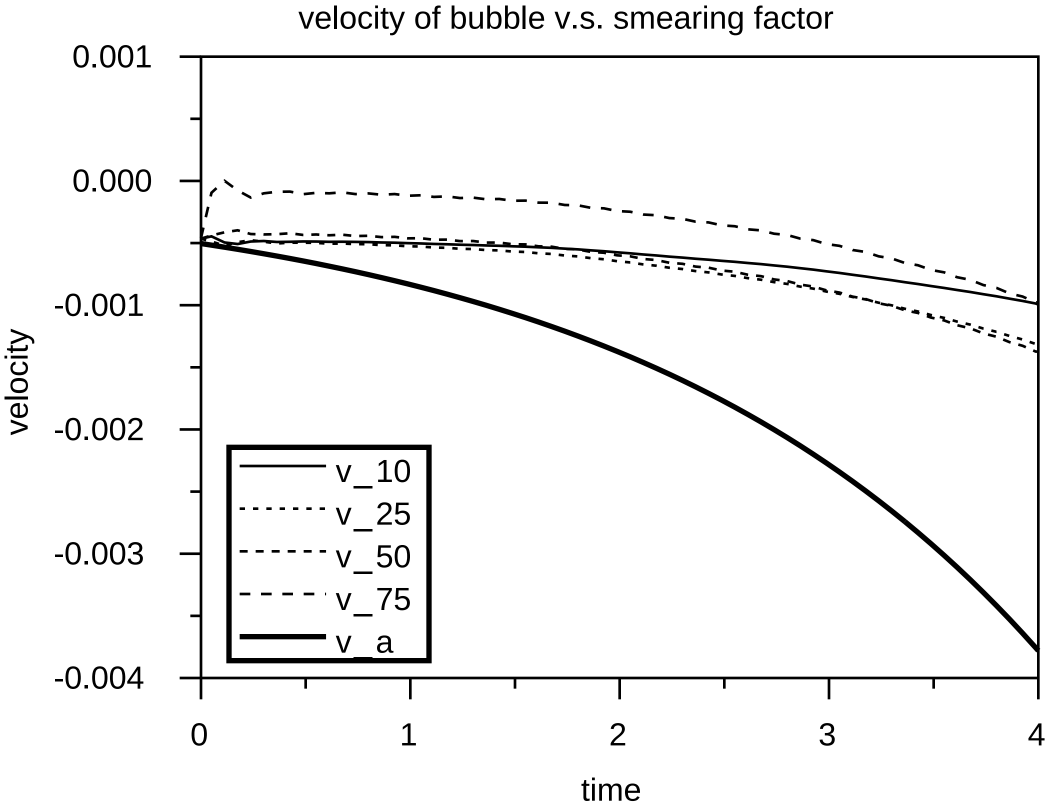
FIGURE 12. Bubble position for various values of the smoothing parameter



In general (when normalized for this round-off error) there is very good agreement with the analytic results to time 1. Even out to time 4, the disagreement is only of order .063% for the smaller values of α. This is in excellent agreement with the 2-D calculations in Gardner et al. [36], and the 3-D calculations of Tryggvason and Unverdi [103], where more complicated front tracking schemes are used to solve the same problem, and the 3-D calculations of Li [65] where a TVD scheme and a level set method are used (also more complicated than the method used in this dissertation.)

Figure 13 sheds more light on the situation. This plot shows the velocity of the bubble for the various values of α. For $\alpha = 10$ the velocity is much too slow compared to the analytic result (again the thick line.) The values 25 and 50 fare only slightly better, but 50 has a slight positive overshoot at the very beginning. This overshoot comes from the generation of the acoustic pulse by the MacCormack methods inability to handle sharp interfaces. The situation is much worse for $\alpha = 75$. Here the overshoot is large, and lasts a long time.

FIGURE 13. Bubble velocity as a function of smoothing parameter



Thus it can be seen that while a certain amount of smoothing in necessary to avoid the acoustic pulse, too much smoothing introduces too much discrepancy between the actual analytic growth rate and the growth rate observed in the calculation. From these calculations, a value of α between 25 and 50 seems to be the best achievable (corresponding to smoothing between 11 and 21 cells.) However, due to the errors mentioned in the previous section, the velocity of the bubble/interface is substantially wrong.

12.2 The Effect of Gravity

The next set of plots explore the effects of gravity on the calculations. The non-dimensional compressibility measures the ratio of the gravitational time scale to the sound speed of the heavy, cold gas:

$$M^2 = \frac{g\lambda}{c^2} \qquad \text{(EQ 10)}$$

As M^2 increases above unity, the possibility of a shock front at the interface becomes more likely. The Lax-Wendroff methods do not handle this situation well. In the following calculations, gravities of 0.125 to 1.0 were tested, which corresponds to values of M^2 from 0.25 to 2.0. The following four plots show the center-line vertical velocity at time 4. "k" is the index for the grid points on the z-axis; k=0 corresponds to the bottom of the domain, and k=256 is the top (gravity points in the +k direction.) The analytic result in these cases should be a V-shaped profile, centered at the middle of the computational domain (where the interface is.) Any deviation from this is an error. In all cases the smoothing parameter is $\alpha = 25$.

In all four figures (Figure 14-Figure 17) there are common features. There is a "wave" near the boundaries. This wave is purely a numerical artifact; it is not part of the analytic solution. It is caused by discretization errors at the boundaries and dispersive errors generating an acoustic pulse at the interface. Because the top and bottom boundaries are rigid fixed lids, the boundaries are reflecting. Thus these errors travel as waves moving at a "numerical" sound speed[9], bouncing back and forth between the walls.

At the lower values of gravity, these waves are small compared to the value of velocity at the interface. But for the larger values of gravity (and thus high values of compressibility) the magnitude of the wave is comparable to the interface. Since this wave is an error, the resultant computation is of very questionable value.

The conclusion which can be drawn from this is that for this code based on the MacCormack method, the values of gravity (i.e., compressibility) must be small. This is in agreement with the known wisdom concerning the Lax-Wendroff methods; the velocities evaluated must be smaller[10] than the local sound speed for the computations to remain valid. Shocks should not be allowed to form.

[9] This numerical sound speed is a function of both the thermodynamical sound speed, and the dispersive error of the MacCormack scheme which results in a phase shift (velocity error) of the various Fourier components of the wave. Effectively, this is just an acoustical wave with a slight variation away for the exact thermodynamical sound speed.

[10] A rule of thumb is to keep the Mach number less than 0.4.

FIGURE 14. Small gravity leads to small disturbances (g = 0.125)

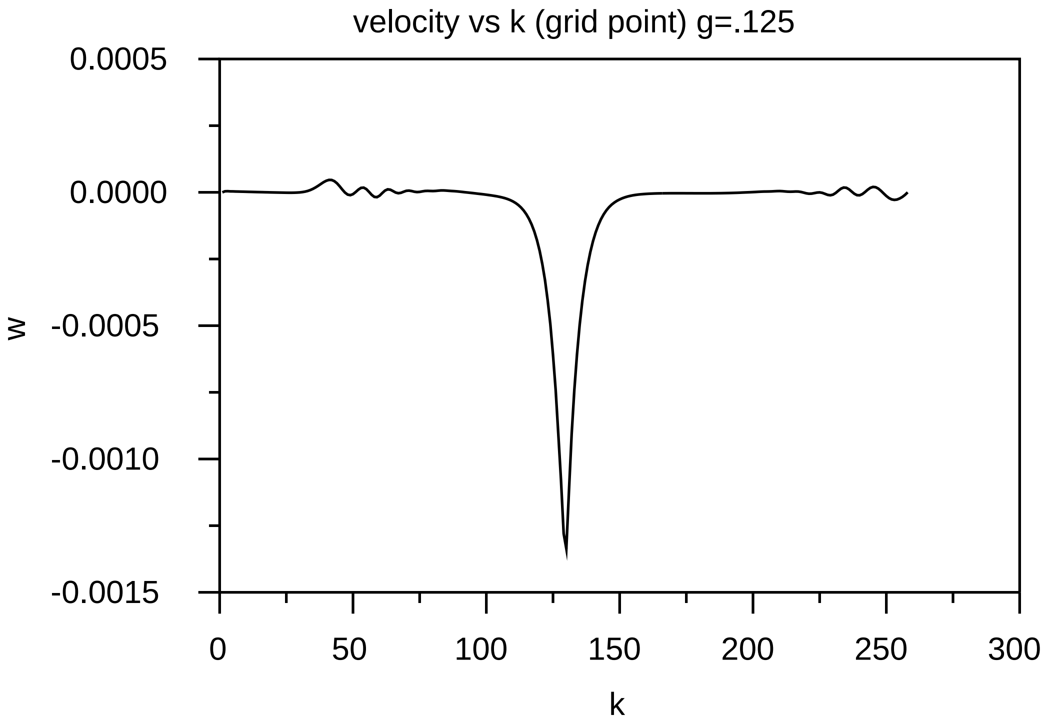


FIGURE 15. A larger value of gravity worsens the situation slightly (g = 0.25)

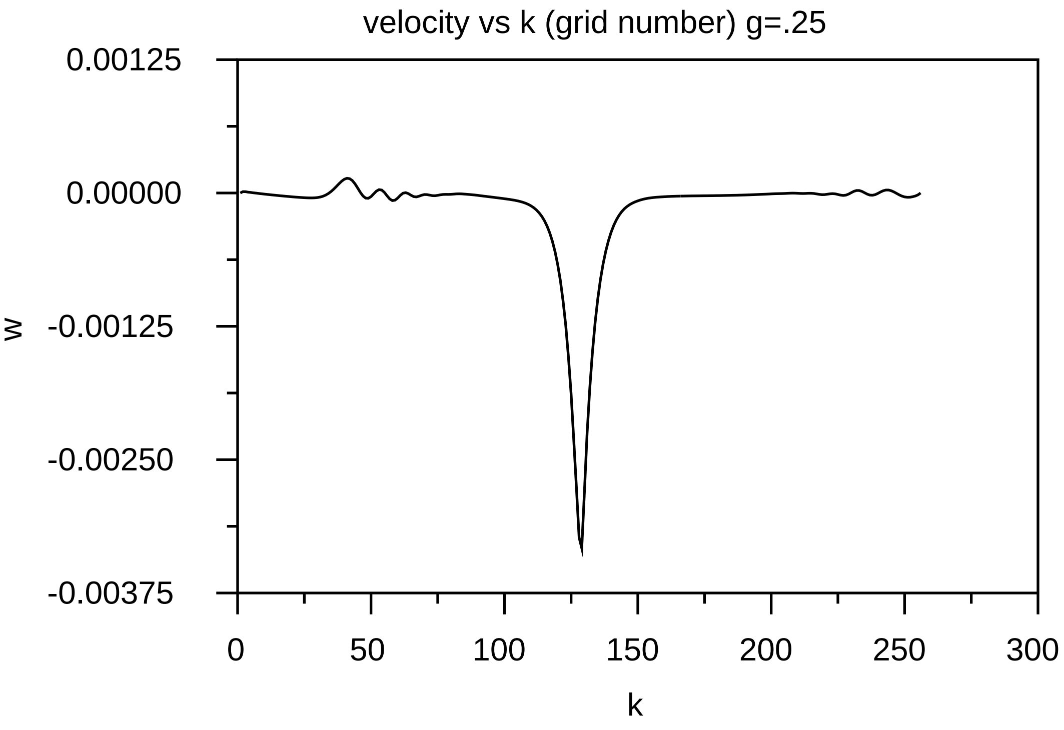


FIGURE 16. The errors are becoming significant (g = 0.5)

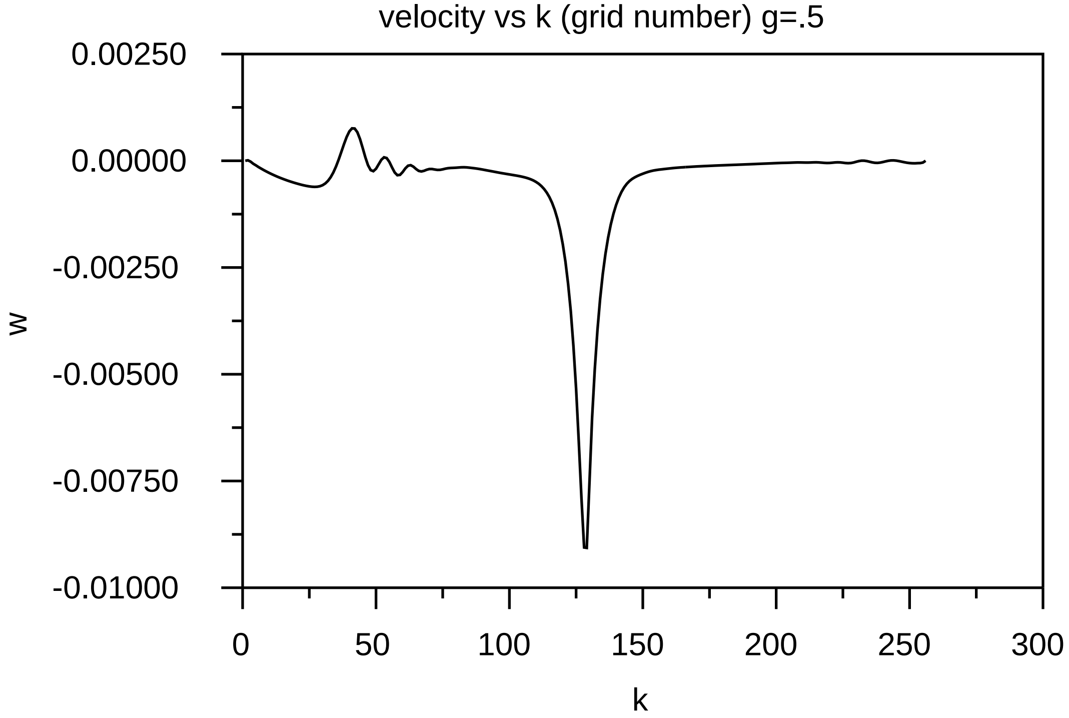
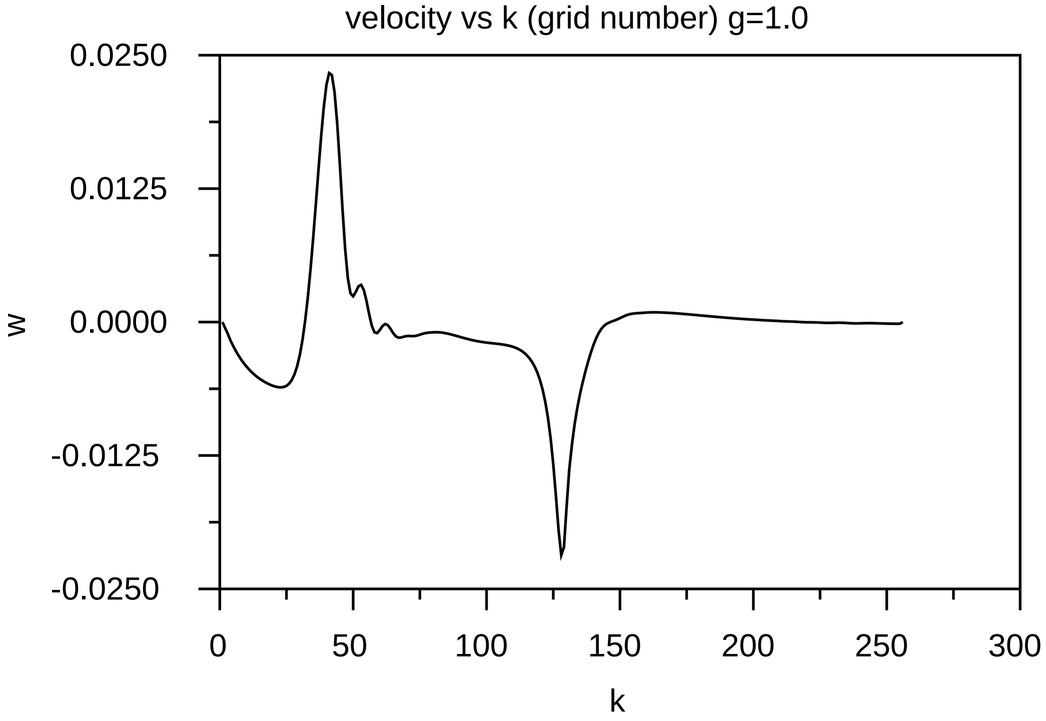


FIGURE 17. The errors are larger than the actual velocity field (g = 1.0)



By these calculations, a maximum value of gravity allowable should be on the order of 0.5, which corresponds to a compressibility of 1.0.

12.3 2-D v.s. 3-D Perturbations

The previous calculations were all for a 2-D perturbation, aligned along the x-axis. The next set of figure presented are for a 3-D perturbation. This perturbation is made by linearly super-imposing a 2-D perturbation aligned along the x-axis with a perturbation aligned along the y-axis. For small enough amplitudes, this superposition should be valid. However, preliminary research by Tryggvason and Unverdi [103] shows that for large enough amplitudes, there can be substantial disagreement between the 2-D and 3-D evolutions of the interface. They state this is caused by the different vortex structures of a 2-D bubble as opposed to a 3-D bubble.

FIGURE 18. 3-D bubble position

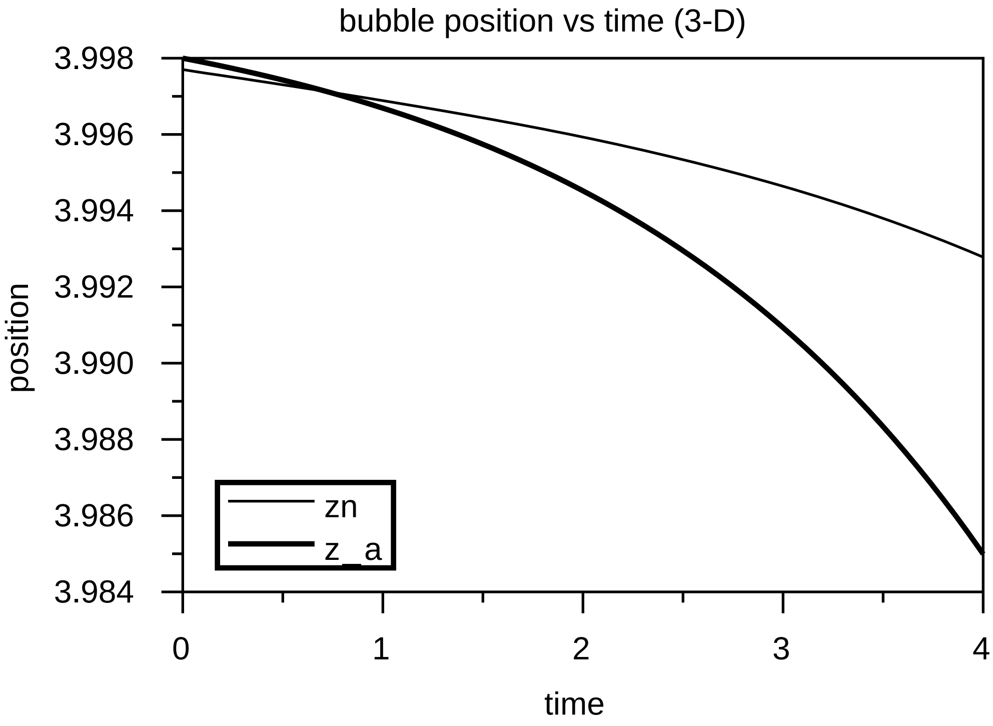


FIGURE 19. 3-D bubble velocity

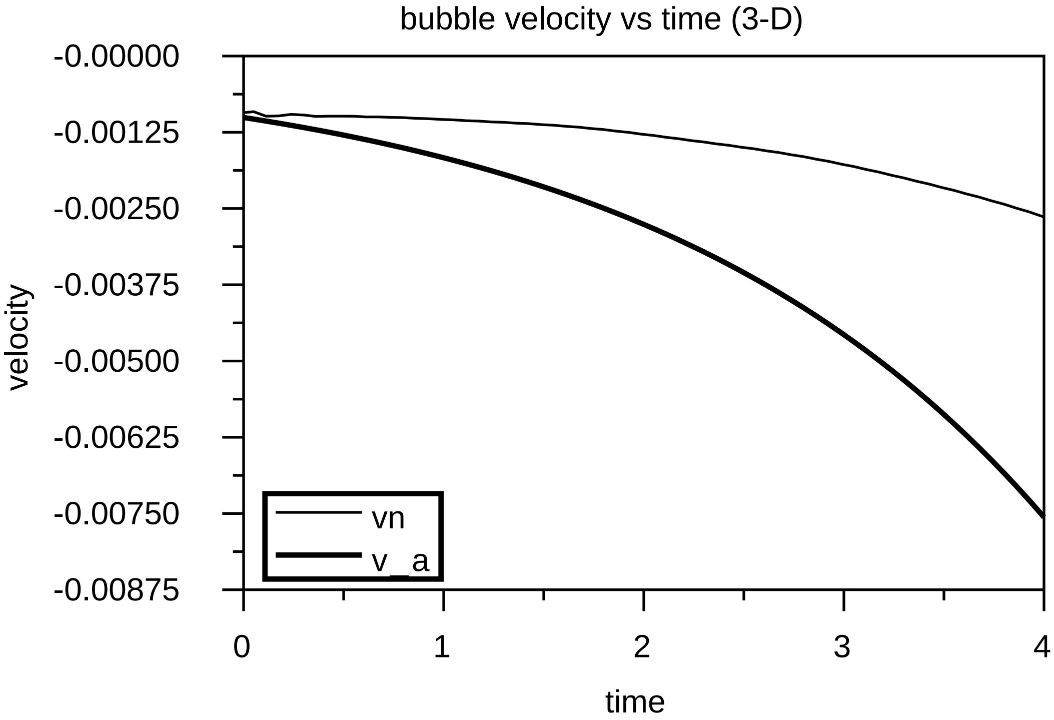


Figure 18 and Figure 19 show the bubble position and velocity for one computational experiment (this calculation has gravity $g = 0.125$, a smoothing parameter of $\alpha = 25$, and the size of the two perturbations are: $\kappa_x = \kappa_y = 2\pi$, $A_x = A_y = 0.001$.) The solid lines represent the analytic solution. The results are comparable to the results of Section 12.1 on page 44. The bubble position tracks the analytic result for a small time, but the velocity is wrong. While the overall behavior is correct, there is a time scale error caused by dispersive errors in calculating the exact velocity.

The growth rate for this calculation, assuming a sharp interface, is calculated from Equation 104 to be $\sigma = 0.5038227$. In this calculation, the actual growth rate is $\sigma = 0.32093$. This is off from the expected value by approximately a factor of 1.57; part of this error is due to the interface no longer being sharp, the other part is due to the dispersive errors of the Lax-Wendroff methods.

12.4 Long Time Calculations

The next set of calculations are to illustrate the long-time behavior of the linear system. The conditions are the same as the calculations in the previous section, except that $g = 0.25$. Certain features are evident.

The linear regime only holds for a short time. As the amplitude of the perturbation increases, the infinitesimal amplitude assumption will no longer be valid. When this occurs, the growth of the perturbation no longer follows the exponential growth rate. The evolution approaches a “free-fall state” where the velocity decelerates. This effect is just starting to become evident in Figure 20 and Figure 21 (especially in the velocity profile where there is a straight ‘free-fall” period, and is just starting to taper off.) These effects occur because of two effects. The first is the vertical boundaries; the lids at the top and bottom do not physically allow for infinite vertical development of the bubbles and spikes.

FIGURE 20. Long time 3-D bubble position

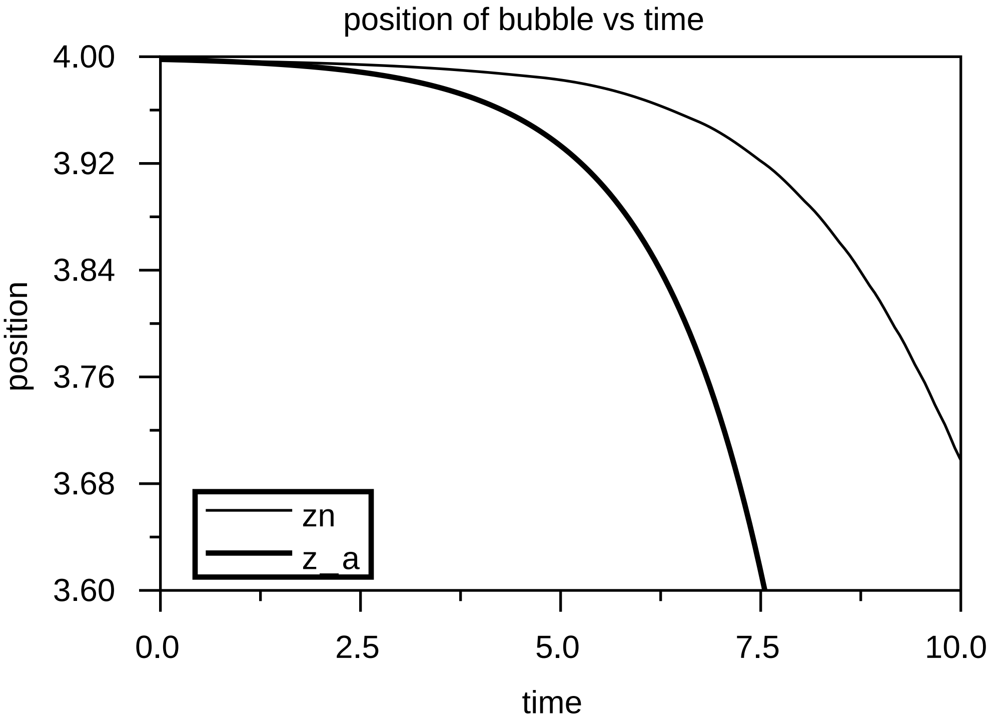
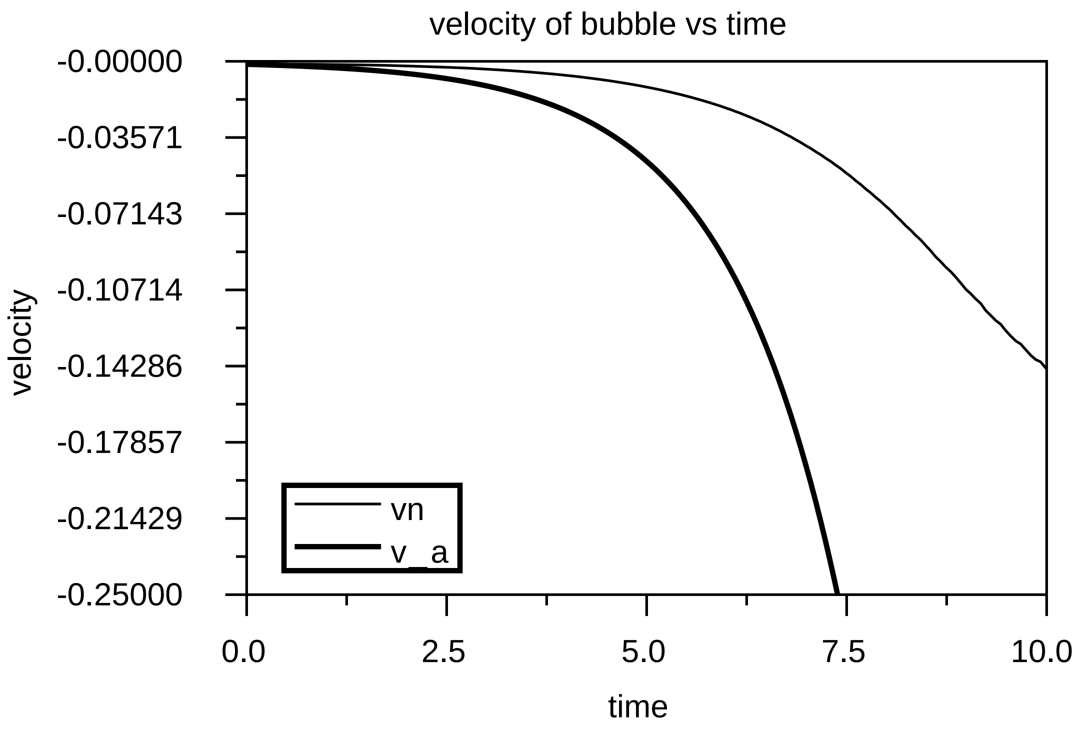


FIGURE 21. Long time 3-D bubble velocity



The second effect is that as the low density bubble rises, it is moving into regions of lower density, and it will begin to float. In calculations taken out to very long times, it is even possible to see an oscillation in the bubble's position as it "bobs" up and down at a level of equal density. (These calculations are not presented here because the errors caused by acoustic waves bouncing off the boundaries make the actual computations have dubious validity.)

FIGURE 22. 0 Isosurface of the passive scalar at time 0

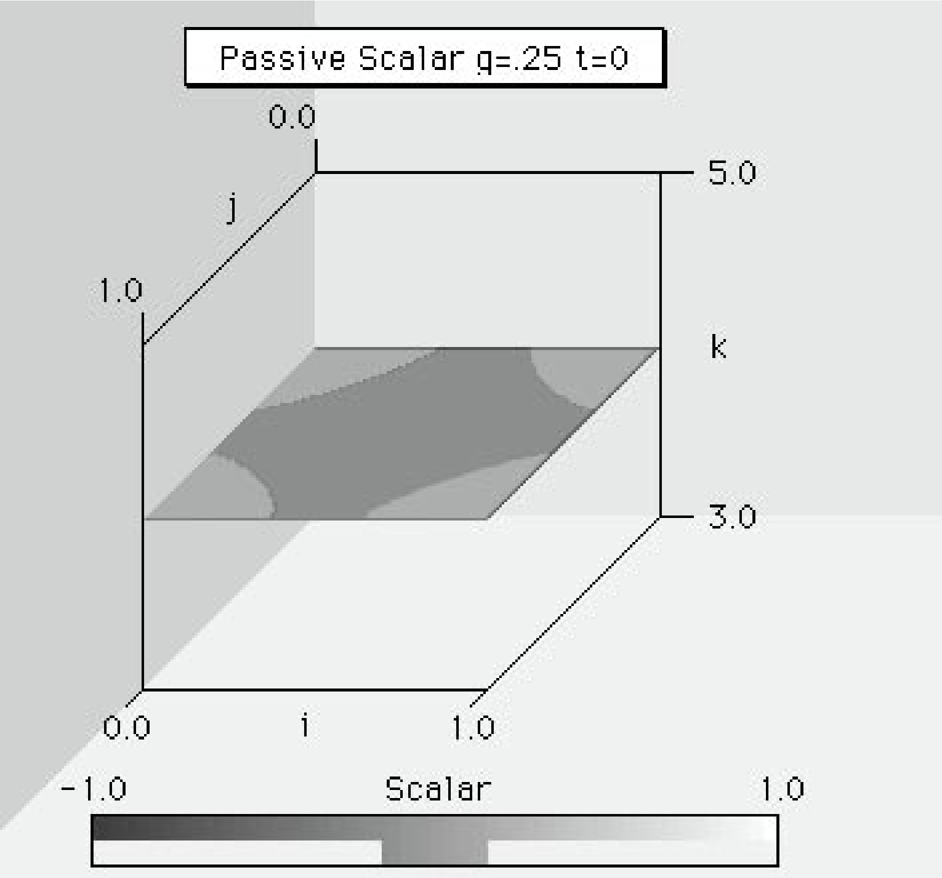


Figure 22 - Figure 24 show the time evolution of the passive scalar for the same computation. The scalar as viewed here is only opaque in the range -0.15 to +0.15, the rest of the range is transparent. This is to aid in viewing the 3-D structure.

Figure 22 is the initial condition. The perturbation is so small that the variation is barely noticeable. The perturbation consists of two super-imposed cosine functions; this gives the initial interface the structure of a slight bulge or bubble.

Figure 23 is taken at time 5. By this time, it is late in the linear regime. The growth still is in the form of a bubble, but the cosine mode in each of the x- and y-axis directions are now beginning to act in a non-linear fashion with other, smaller scale modes. The amplitude is still quite small however. As an aside, the slight gaps in the figure come from the form of interpolation of data that the graphing program (SpyGlass Dicer on a Macintosh Centris 650) utilizes. It is not an error.

FIGURE 23. 0 Isosurface of the passive scalar at time 5

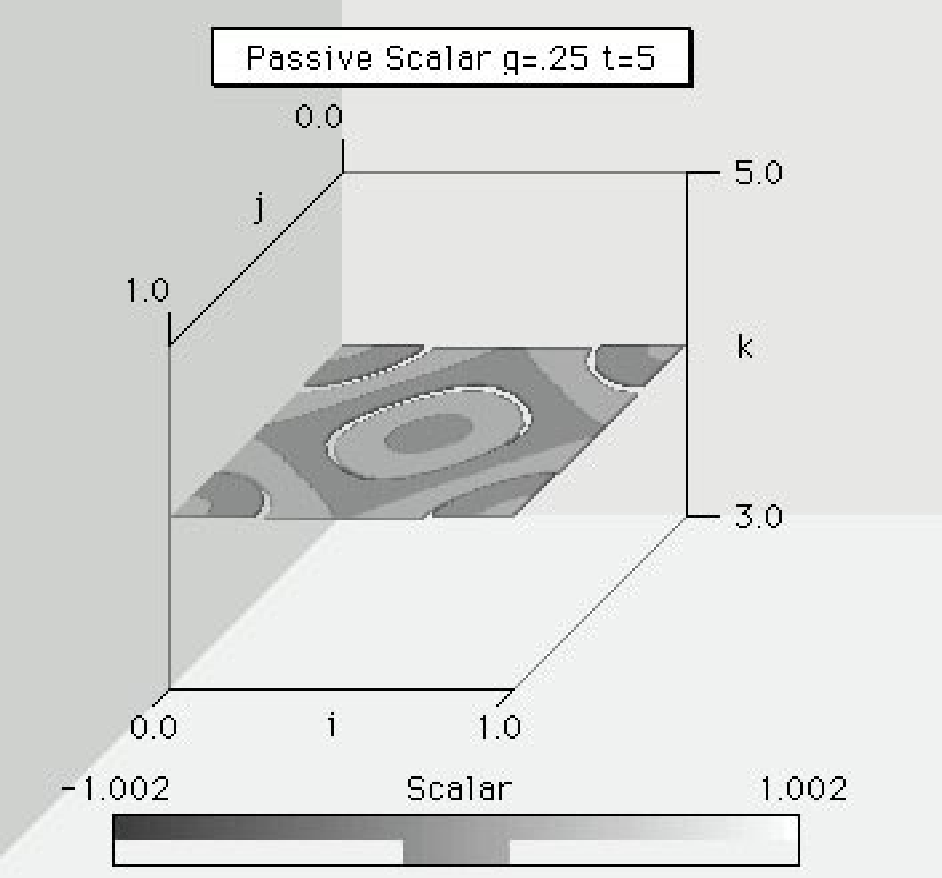
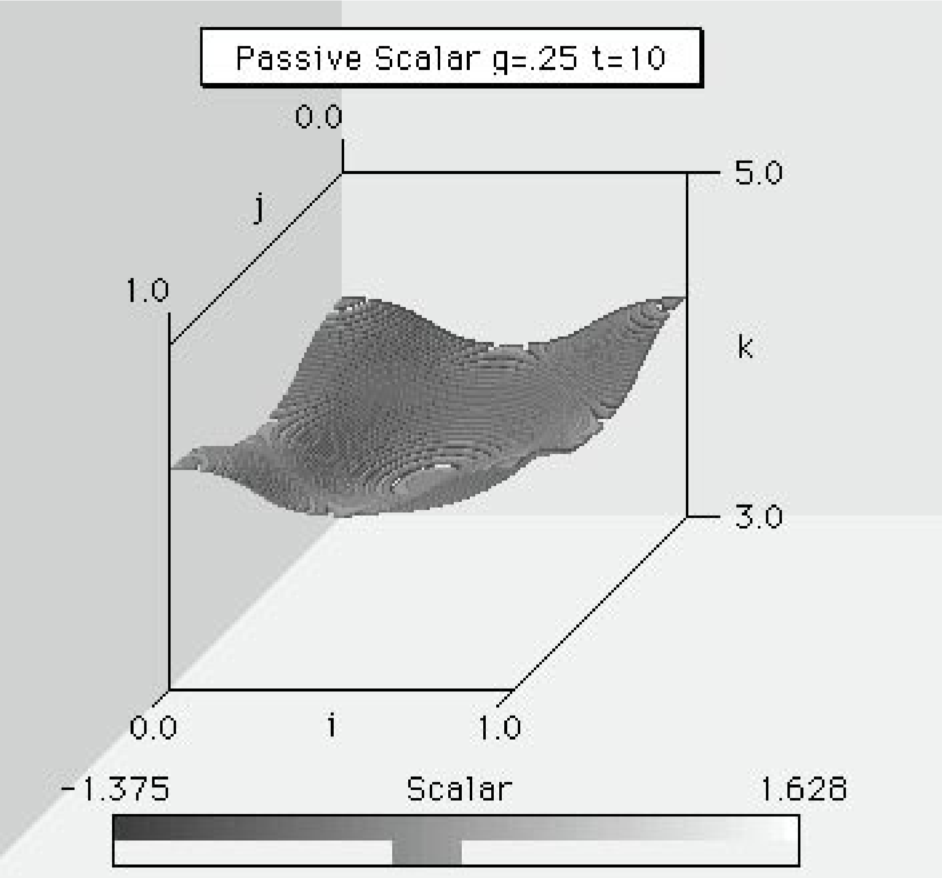


Figure 24 is the situation at time 10. The amplitude of the perturbation is substantially larger than in the previous figure. At this stage, the growth is almost entirely out of the linear (i.e., exponential) regime. Non-linear mechanisms control the expansion of the bubble. This is evident by the velocity evolution in Figure 21; the velocity is no longer increasing in an exponential manner.

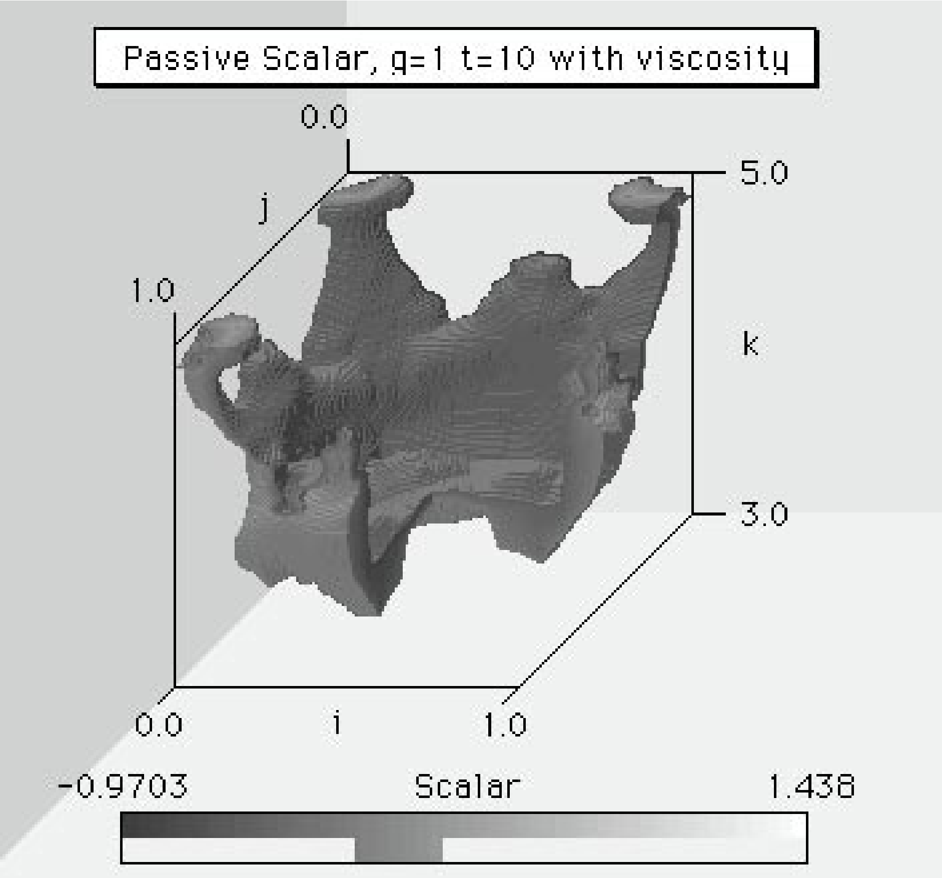
FIGURE 24. 0 Isosurface of the passive scalar at time 10



The final 3-D image presented is from a different computation. In Figure 25, the gravity is now set to unity. This increases the compressibility, and thus the time scale. The computation is run with eddy-viscosity to help maintain stability. The plot is presented to show the non-linear effects in a very advanced stage (a wedge has been removed to help in visualization of the complex form.) The perturbation no longer appears as a bubble. The edges are starting to roll up. The action of viscosity has caused a thickening of the interface. Acoustic waves and other numerical errors have broken the symmetry. Notice the classic non-linear behavior of the Rayleigh-Taylor instability. The bubble of light gas is broadening downwards against gravity (which is up,) while thin spikes[11] of heavy gas shot in the direction of gravity (up.)

[11] at the corners - don’t forget the domain is periodic in the horizontal

FIGURE 25. 0 Isosurface at time 10 with viscosity and a larger value of gravity



12.5 The Effect of Viscosity

All of the previous calculations contained no viscosity terms, other than numerical truncation error terms of the fourth-order (hyper-viscosity.) This results in the interface remaining sharp (up to the smoothing parameter) after a substantial amount of time. Other than convective mixing, there is no mechanism for density, entropy, the passive scalar to mix or diffuse. However, because of the lack of viscosity, the non-physical erroneous acoustic pulses are free to travel about. These result in undesirable dispersive errors.

The following computation illustrates this. This computation was run twice, once without and once with the eddy-viscosity term turned on. The Smagorinsky eddy-viscosity used here is described fully in Section 22.1 on page 101. The parameters are as follows: $\alpha = 25$, $g = 0.25$. The computation is run out to time 10. The perturbation is 3-D.

FIGURE 26. Initial Conditions

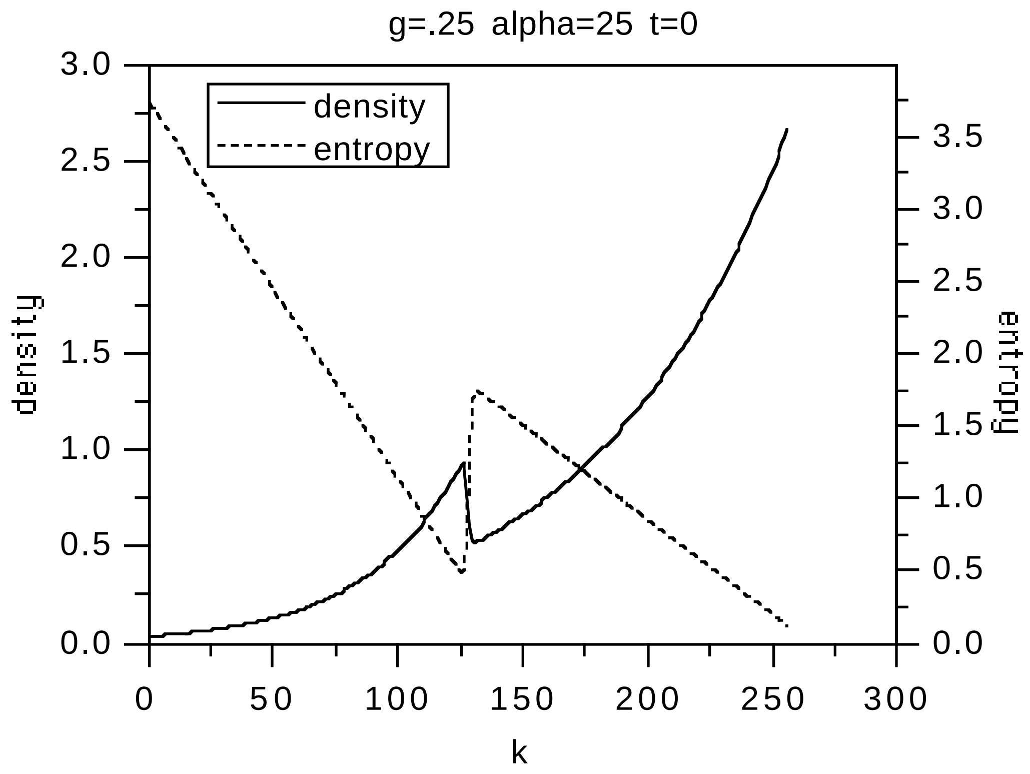


FIGURE 27. t=10 No Viscosity

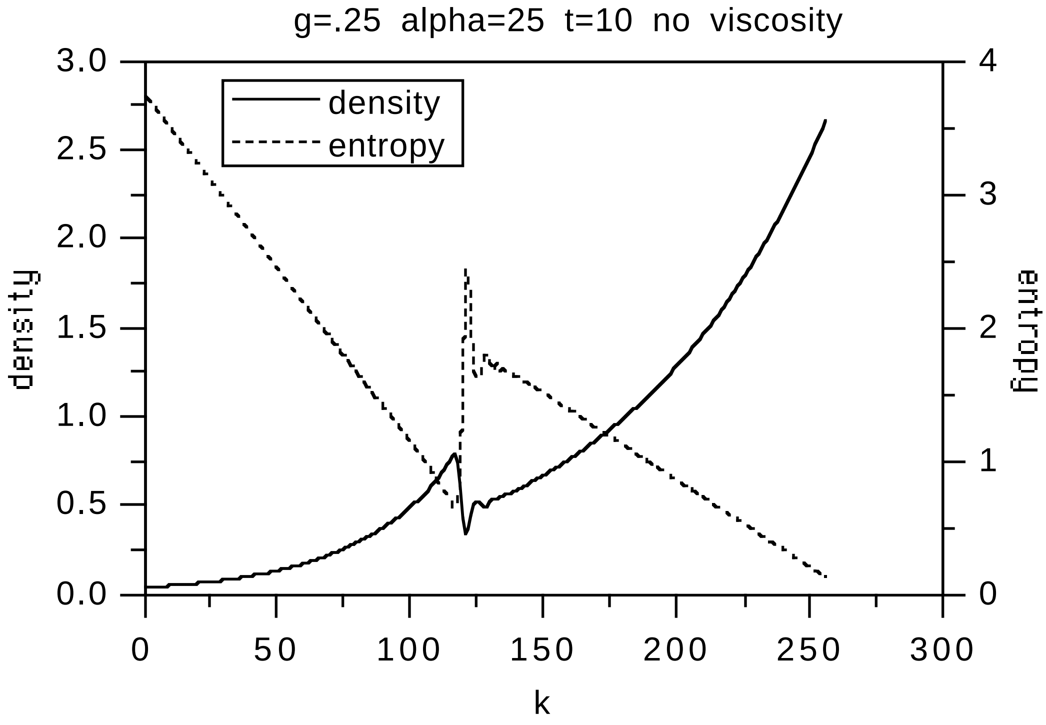


FIGURE 28. t=10 Viscosity

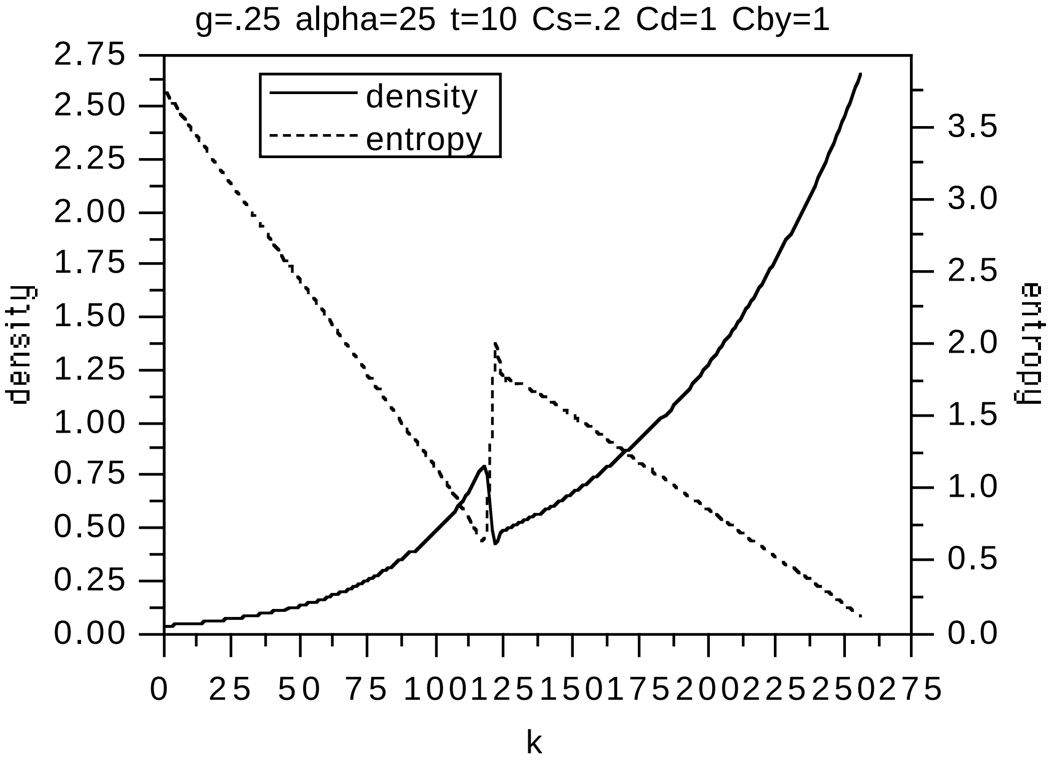


Figure 26 shows the initial condition. The interfaces are fairly sharp (about 21 zones.) Figure 27 shows the situation at time 10. The position of the interface has not changed. This is in agreement with the analytic model. But there is a "overshoot" and "undershoot" at the interface. This is non-physical. As mentioned earlier, this is due to Lax-Wendroff methods' inability to handle sharp interfaces well; dispersive errors of this type result.

Figure 28 shows the same computation with the eddy-viscosity turned on. In this case, the constants of the eddy-viscosity terms[12] are set to $C_S = 0.2$, $C_D = 1.0$, and $C_{By} = 1.0$. As can be seen, this viscosity is enough to reduce the overshoots and undershoots substantially. However, it is not so great that it caused an undue amount of diffusion of the density or entropy. The interface is now spread out over approximately 33 zones, for an increase (i.e., diffusion) of about 12 zones. There is a cost in reducing the overshoots though. The calculation with no viscosity took 800 time steps. The calculations with the eddy-viscosity took 2400 time steps, for an increase by a factor of three. This is due to no longer being

[12] Defined in Section 22.1 on page 101

able to solely use the CFL condition for stability. With the addition of a viscosity term, it is necessary to take into account the cell Reynold’s number criteria:

$$Re_{\text{cell}} = \frac{c_H \Delta x}{\nu_T} \le 2 \qquad \text{(EQ 11)}$$

Strictly speaking, this is not a stability condition. It is actually a question of resolution; in order to correctly resolve the flow, and to dampen small scale oscillatory numerical errors, it is necessary to satisfy the cell Reynold’s number criteria. With the inclusion of the turbulent eddy-viscosity, it is necessary to effectively reduce the cell size, which in turn reduces the time step size by the CFL condition. Both Strikwerda [99] and Oran and Boris [78] give good expositions on the cell Reynold’s number criteria. Strikwerda also has detailed mathematical analysis for the linear advection-diffusion problem.

13.0 The Random Mixing Layer

13.1 Introduction

The final problem addressed in this dissertation is that of the randomly perturbed interface. In the last section, the fluid was perturbed by a single sine mode. It would of course be possible to superimpose many modes together; this would result in initially a number of bubbles and spikes of various radii growing. Eventually, one mode (the dominant mode) wins out, and effectively “swallows” up the others. This occurs in the non-linear regime of the growth of the perturbations. Several of the previously cited computational works examine the multiple mode problem.

While examining these multiple modes is of interest (mainly due to the presence of an analytic solution in the linear regime to check against,) the real interest with multiple modes is what happens to a real fluid. In a real fluid, it is impossible to have just one mode. In the experimental set-up, even if the box is cleverly shaken to develop a single standing wave, there will be many sub-harmonic waves present. Even when the interface is perfectly flat, there are molecular fluctuations and microscopic vibrations that will result in perturbations. One way to model this effect is the superposition of multiple, linear modes.

These very small, possibly non-resolvable, perturbations can also be thought of as randomly occurring. One way to numerically accomplish this is by randomly assigning amplitudes, wavenumbers, and phases to a specified number of modes. This is the method used in Youngs [113] and Li [65]. This method gives good results, and it is possible to “deconstruct” an analytic answer using the known initial conditions. However, the statistics are limited since typically a small number of modes, etc. are used.

It is easy to believe that the random fluctuations caused by vibrations, thermal fluctuations, etc. should behave statistically as white noise. The above method of randomly perturbing the initial conditions falls far short of this. A method will be presented here which will put a statistically white perturbation on the stationary, unstable initial conditions of the Rayleigh-Taylor problem. This method is an extension of the Smagorinsky eddy-viscosity model used in the previous section to dampen the unwanted acoustic waves present due to the sharp density discontinuity and slightly incorrect boundary conditions. A stochastic backscatter term is added to this eddy-viscosity model to account for the back scattering of kinetic energy from the unresolved subgrid-scale region to the resolved length scales of the computational grid. “Appendix B - Turbulence Modelling” gives a complete definition of this stochastic backscatter term as developed in the works of Leith [58][59].

Thus, for this dissertation, the stochastic backscatter model can do double duty. First, it is a proven extension to the Smagorinsky eddy-viscosity model. Secondly, it can be used to add the random forcings (i.e., perturbations) to the stationary, unstable initial conditions of the Rayleigh-Taylor problem.

13.2 Properties of Random Perturbations

As previously mentioned, Youngs and Read [87][112][113] performed experiments which looked at the randomly perturbed interface. Using dimensional analysis, they showed that the growth of the mixing layer of this problem should be:

$$h_{av}(t) = \xi A g t^2 \qquad \text{(EQ 12)}$$

where A is the Atwood number, g is gravity, and t is time. $h_{av}(t)$ is the average thickness of the mixing layer in the heavy fluid's half plane (i.e., average penetration of the bubble,) measured at a mixing ratio of 95%. ξ is an experimentally determined dimensionless mixing coefficient, found to be fairly independent of both A and g. In their experimental work, they determine are range of $\xi \approx 0.06$ to $\xi \approx 0.07$, but in their two-dimensional computational work they get $\xi \approx 0.04$. Li [65] found an average value of $\xi \approx 0.071$ in his three-dimensional computational work.

Youngs [112] gives a brief outline for the development of Equation 12. He states that there are three basic stages to the mixing process of a randomly perturbed interface. In the first stage, a small wavelength from among those present will be dominant (Chandrasekhar [20].) This grows at an exponential rate, but soon saturates, and its growth rate slows down. This is stage two, where non-linear interactions dominate the development, and large scale structures form. All memory of the initial conditions is lost. This leads to stage three. With all memory of the initial conditions lost, by dimensional reasoning the only length scale left is from gt^2. The dependence on the density jump (or Atwood number A) can also be deduced from Chandrasekhar [20], since the growth rates of any perturbation (regardless of amplitude or wavelength!) depends on the ratio of densities. This is also evident by examining Equation 104.

14.0 The Random Mixing Layer Calculations

The following calculations will attempt to prove several different points:

1. The stochastic backscatter terms initiates random mixing.
2. This mixing closely fits the expected growth rates as found in physical experiments.
3. The constant ξ is invariant in the regimes tested.
4. This correct behavior is independent of the smoothing parameter, and the backscatter constant C_B.

The flow regimes tested will be in ranges for which the MacCormack scheme handled the linear perturbation problem well. This includes gravity values from g = 0.125 to

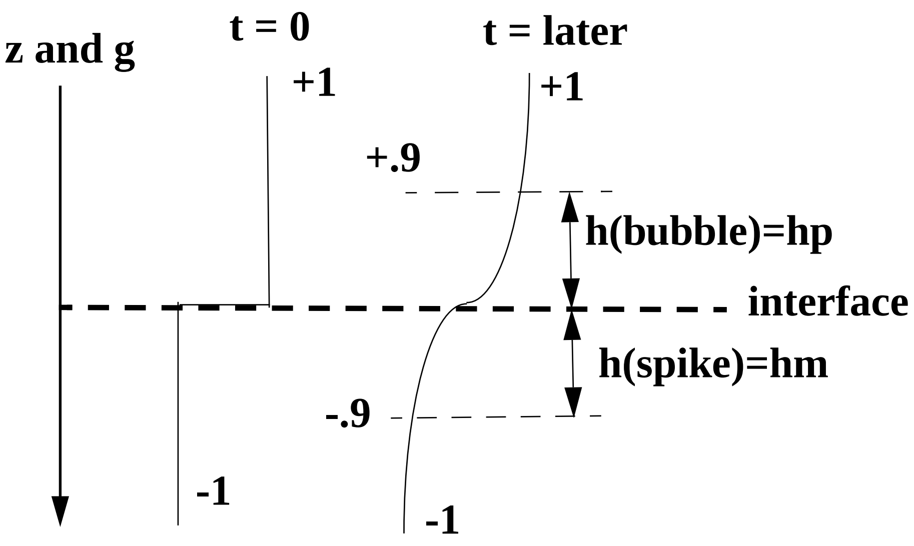
$g = 0.5$, and smoothing parameters in range $\alpha = 25$ to $\alpha = 50$. The Atwood number for all computations presented here is $A = \frac{1}{3}$. The computational domain is $1 \times 1 \times 8$, with a computational grid of $32 \times 32 \times 256$. The base scale gravity and densities are unity ($g_z = 1$, $\rho_0 = 1$,) as is the base velocity scale which is set by the hot gas sound speed (${c_H}^2 = 1$.)

A passive scalar is used to determine the thickness of the mixing layer. There is no scalar diffusivity in this model, so any given point in space can contain only one fluid or the other. If the scalar is positive, the fluid is considered to be fluid A (initially in the half plane from z = 0 to z = interface.) If the scalar is negative, the fluid is fluid B (initially in the other half plane.) To determine the average value of the scalar as a function of z, the sign of the scalar is averaged across the horizontal x-y planes. This effectively defines a mixing ratio χ with values in the range -1 to +1, where this mixing ratio is the forementioned average:

$$\chi(z, t) = \mathrm{ave}_{x, y}(\mathrm{sgn}\,\Phi(x, y, z, t)) \qquad \text{(EQ 13)}$$

At time zero, this results in a step function which is +1 in the half plane from 0 to the interface, and -1 in the other. Thus, even though the scalar itself has a continuous hyperbolic tangent profile (by the smoothing parameter,) the mixing ratio starts out sharp. As the perturbations progress, the mixing ratio as a function of z will start to smooth out due to random mixing. The edges of the mixing zone are defined as the region where the fluid is 95% purely one fluid or the other; in this case that corresponds to a mixing ration of $\chi = 0.9$ to $\chi = -0.9$. The fluid in the range $\chi \in (-0.9, 0.9)$ is considered to be “fully” mixed, fluid in the ranges $\chi < -0.9$ and $\chi > 0.9$ is considered non-mixed. Figure 29 should illustrate these definitions, where h(bubble) gives the average mixing depth of the bubble, etc.

FIGURE 29. Mixing Ratio Function



In the following graphs “hp” is the mixing depth of the bubble’s penetration into the heavier fluid, and “hm” is the spikes penetration into the lighter fluid. “hp” is equivalent to the function $h_{av}(t)$.

14.1 The Invariance of ξ

This first set of computations prove the first three points from above. The three-dimensional Navier Stokes equations of Section 17.4 on page 86 including the Smagorinsky eddy-viscosity of Section 22.1 on page 101 and the stochastic backscatter of Section 22.2 on page 106 are integrated using the two-step MacCormack method of Section 24.2 on page 114. The initial conditions are the smoothed interface hydrostatic state of Section 11.0 on page 33. A number of computations were done in order to evaluate ξ for various values of gravity and the smoothing parameter. The backscatter adjustable coefficient is set to $C_B = 0.2$ in order to be in agreement with the values used by Leith [58][59], and to have the same value as the Smagorinsky coefficient (which is also set to 0.2)

The results of these computations are summarized in Table 14 . The values are in close agreement with Youngs [112] and Li [65]. The values for $g = 0.5$ are the worse results, but these calculations (as seen in the previous sections) are at the limit of the capabilities of the MacCormack scheme for problems of this sort.

There are two methods to recover the value of ξ for the data. The first method is by fitting a second order polynomial in t to the data. This is illustrated by the curve fit in Figure 30. The second method corresponds to the values in Table 14 which comes from normalizing the data by dividing out the Atwood number, gravity, and t^2 , i.e. by plotting,

TABLE 14. Values of ξ calculated by normalizing against t^2

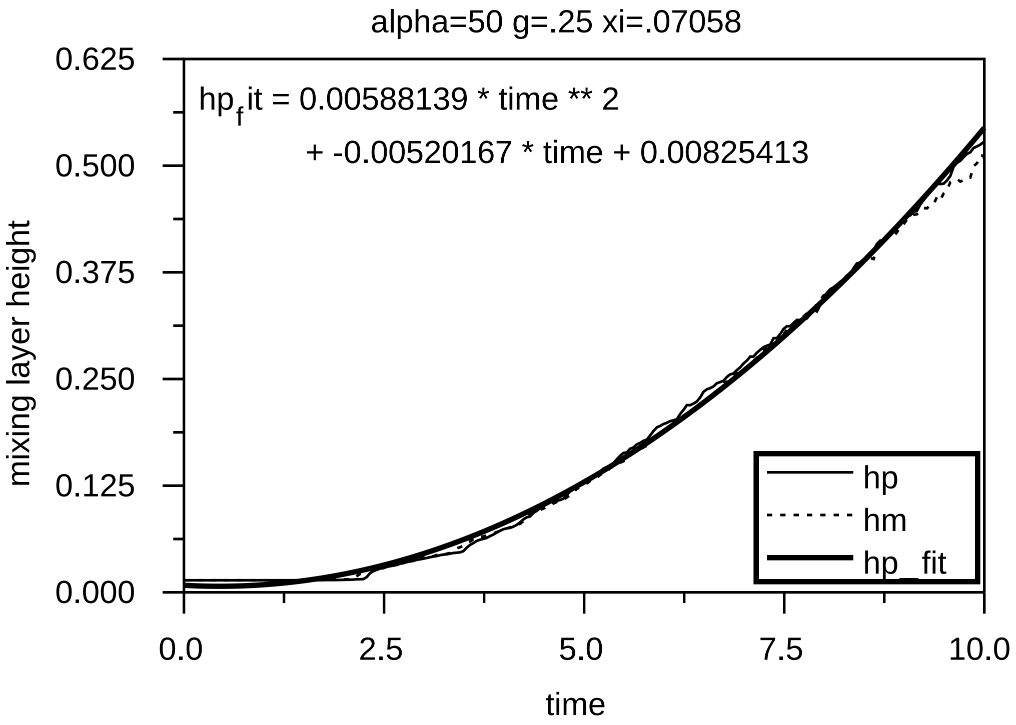
ξ	$\alpha = 25$	$\alpha = 50$
$g = 0.125$	0.0600	0.0625
$g = 0.25$	0.0575	0.0626
$g = 0.5$	0.0520	0.0550

$$\xi = \frac{h_p}{gAt^2} \qquad \text{(EQ 14)}$$

as a function of time. This is illustrated by Figure 31. This method is the better of the two in that the normalization with respect to t^2 is exactly accurate, whereas the curve fit is only an estimate. Also, whenever a figure like Figure 31 is flat, that portion of the function is definitely a function of t^2 . Sometimes figures (and curve fits) like Figure 30 can be misleading; a third order polynomial probably would fit the curve as well, but be in violation of the dimensional analysis! The values in Table 14 are very close to the expected value of 0.06; however, the values for g = 0.5 are still not as accurate.

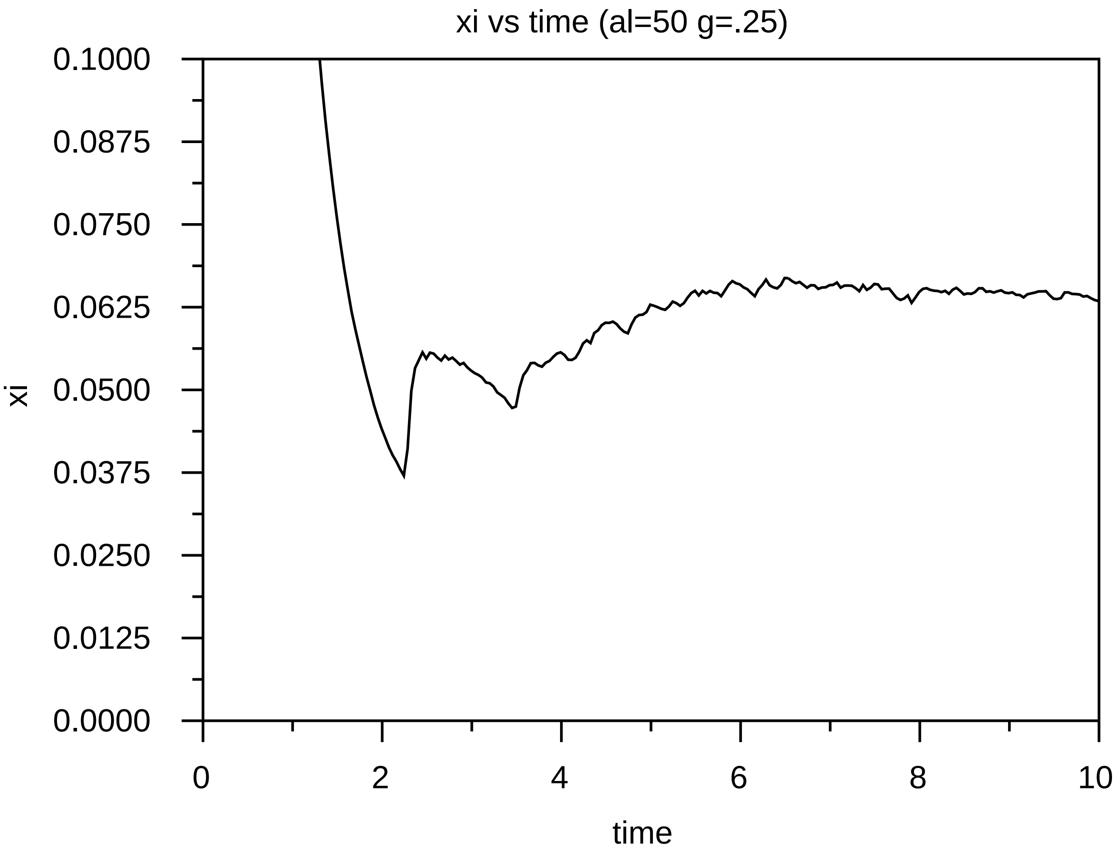
Figure 30 is a representative plot of the thickness of the mixing layer versus time for $\alpha = 50$ and $g = 0.25$. The plots of hp and hm starts out flat until time 2.5. This is due to the finite thickness of the computational grid; the flat portion corresponds exactly to a half of 90% of a grid zone Δz. Any mixing layer which is narrower than this thickness is unresolvable. Thus the mixing layer must become thicker than this value to be seen. A second order polynomial fit is made to the hp curve. This indicates that $\xi = 0.07058$

FIGURE 30. Plot of spike and bubble mixing depth as a function of time



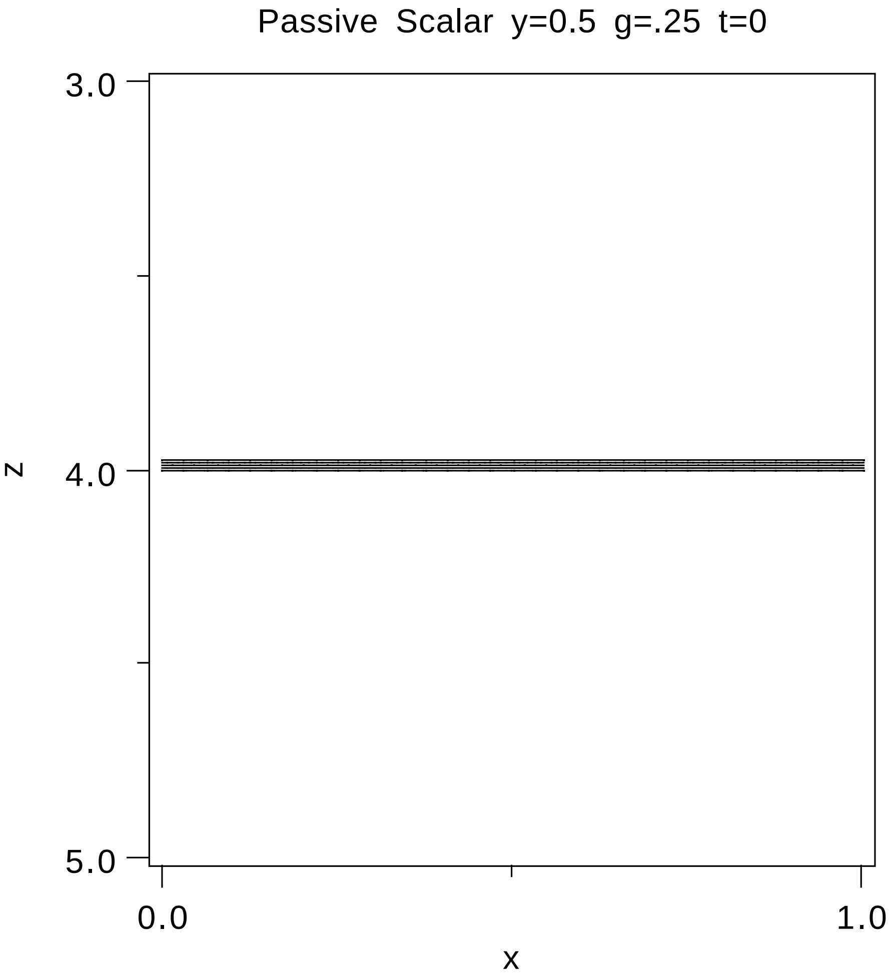
The next plot, Figure 31, is the same data set, but is solely a plot of ξ; the Atwood number, gravity, and t^2 values have all been divided out. This plot proves that after a starting transient, that the growth of the mixing layer is truly a function of t^2, and that ξ asymptotes to a value of approximately 0.0626. This value is slightly better than the value from the second order polynomial fit of Figure 30, in that it is closer to the desired range of 0.06 to 0.07. The starting transient is due to the lack of resolution below the grid size Δz, as previously mentioned.

FIGURE 31. The evolution of ξ



The final group of figures (Figure 32 through Figure 37) show the evolution of this random mixing layer from t=0 to t=10 (for the same parameters as above.) There are two figures for each time step shown. One figure is a contour plot which shows an x-z plane sliced at a y value of 0.5 (centerline.) The other figure is a 3-D image at the same time value. This shows a side plane, and the back plane of the data set. In both cases, the z-axis is only shown between the values of 3 and 5, as all pertinent detail occurs in this range. Between these two types of images, it is clear that the initially flat (and reasonably flat) interface has its symmetry broken, and motion is initiated by the random fluctuations from the stochastic backscatter term.

FIGURE 32. Initial conditions, no observable mixing (contour plot)



Note that in the 3-D images, the scalar does not stay entirely in the range -1 to +1, which should be the case given conservation principles. There are two reasons for this. The first reason is that there still exists small errors in the velocity field calculations (acoustical type errors.) Even with the eddy-viscosity terms, there will still be overshoots and undershoots of the velocities. This results in passing on these errors to the passive scalar. The other cause of the error is the 3-D imaging program’s data interpolation routines and resolution.[13] Of these two errors, the first is the more dominant.

[13] Spyglass Dicer on a Macintosh Centris 650

FIGURE 33. Initial conditions (3-D view)

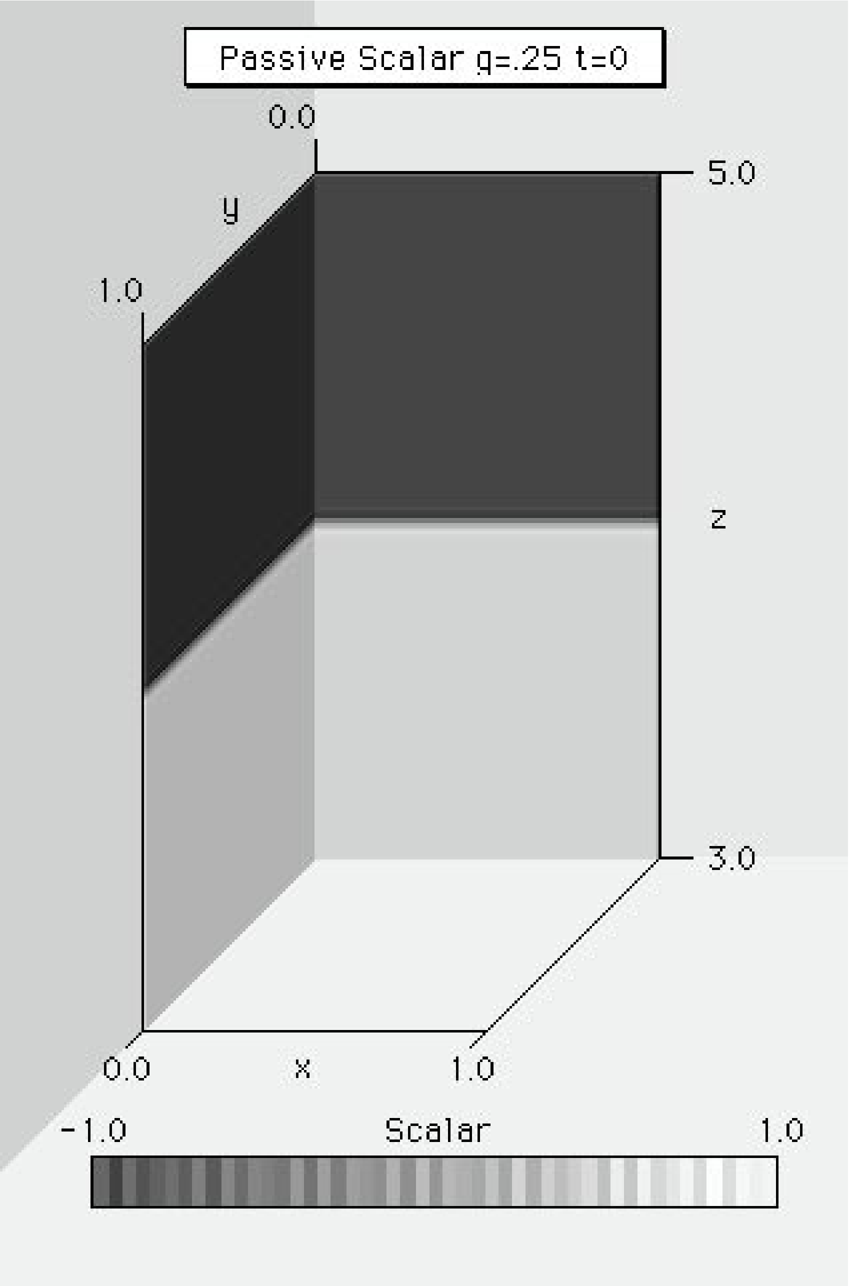
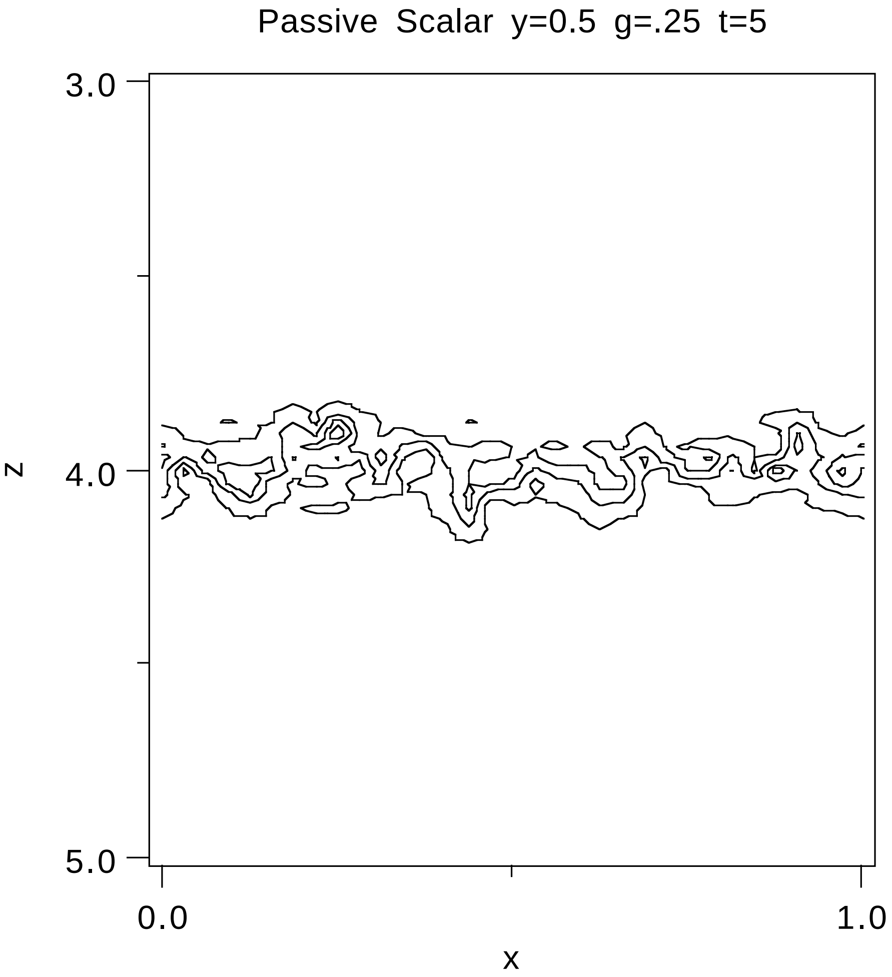


FIGURE 34. Intermediate condition (contour plot)



At t=5, there are still observable large-scale structures, with a wavelength of perhaps a third. There are also numerous other modes superimposed on top of this dominant structure. Bubbles and spikes are evident. Since the density jump is only a factor of two, the shapes of the bubbles and spikes are similar.

FIGURE 35. Intermediate condition (3-D view)

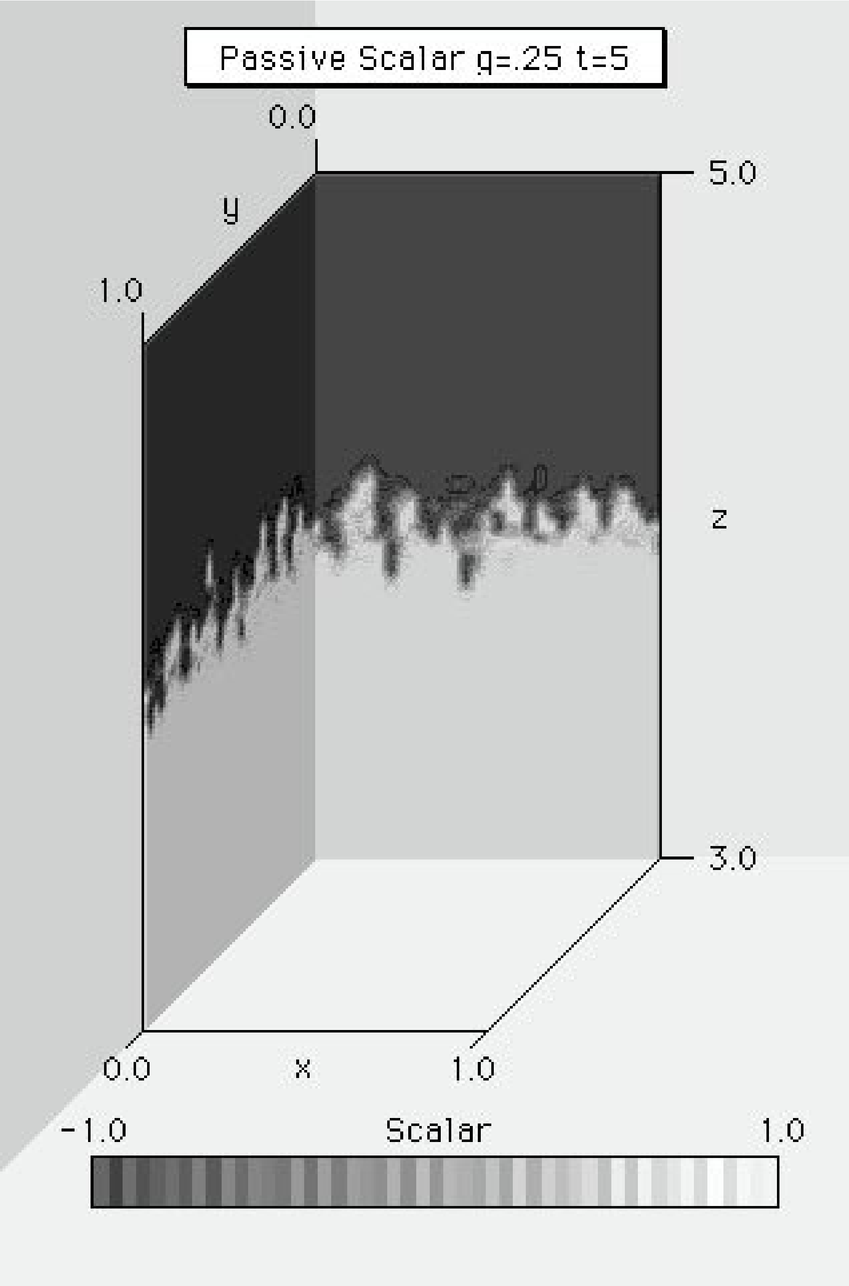
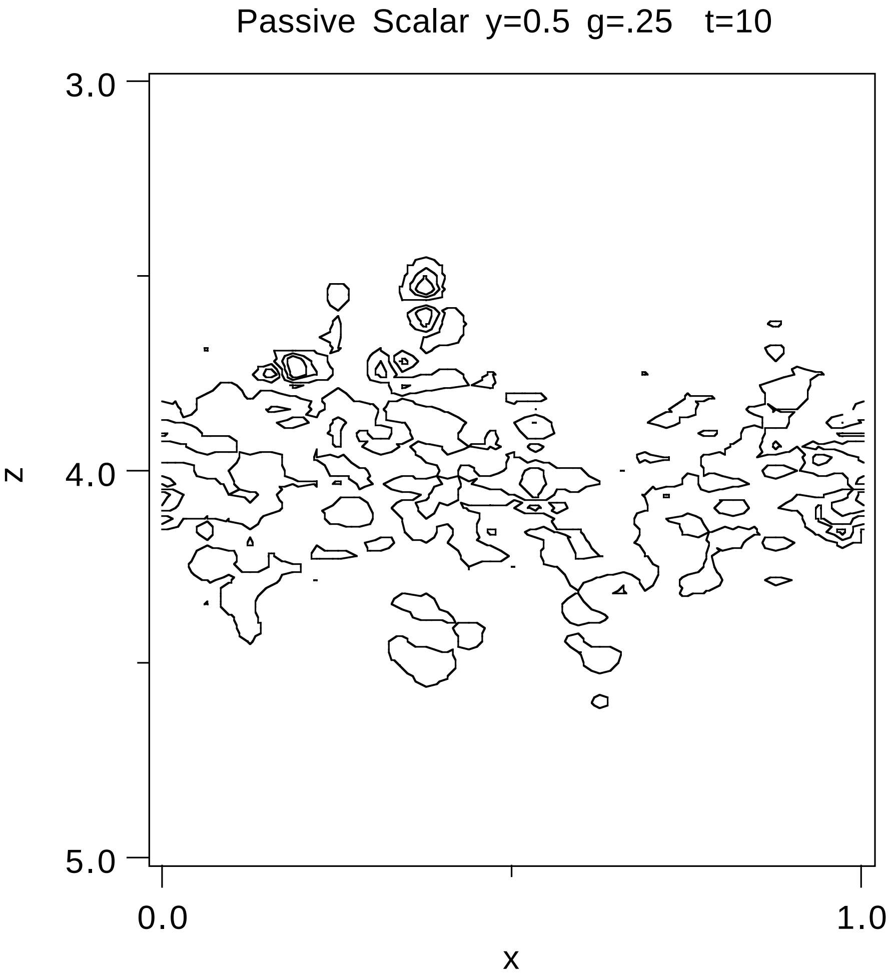
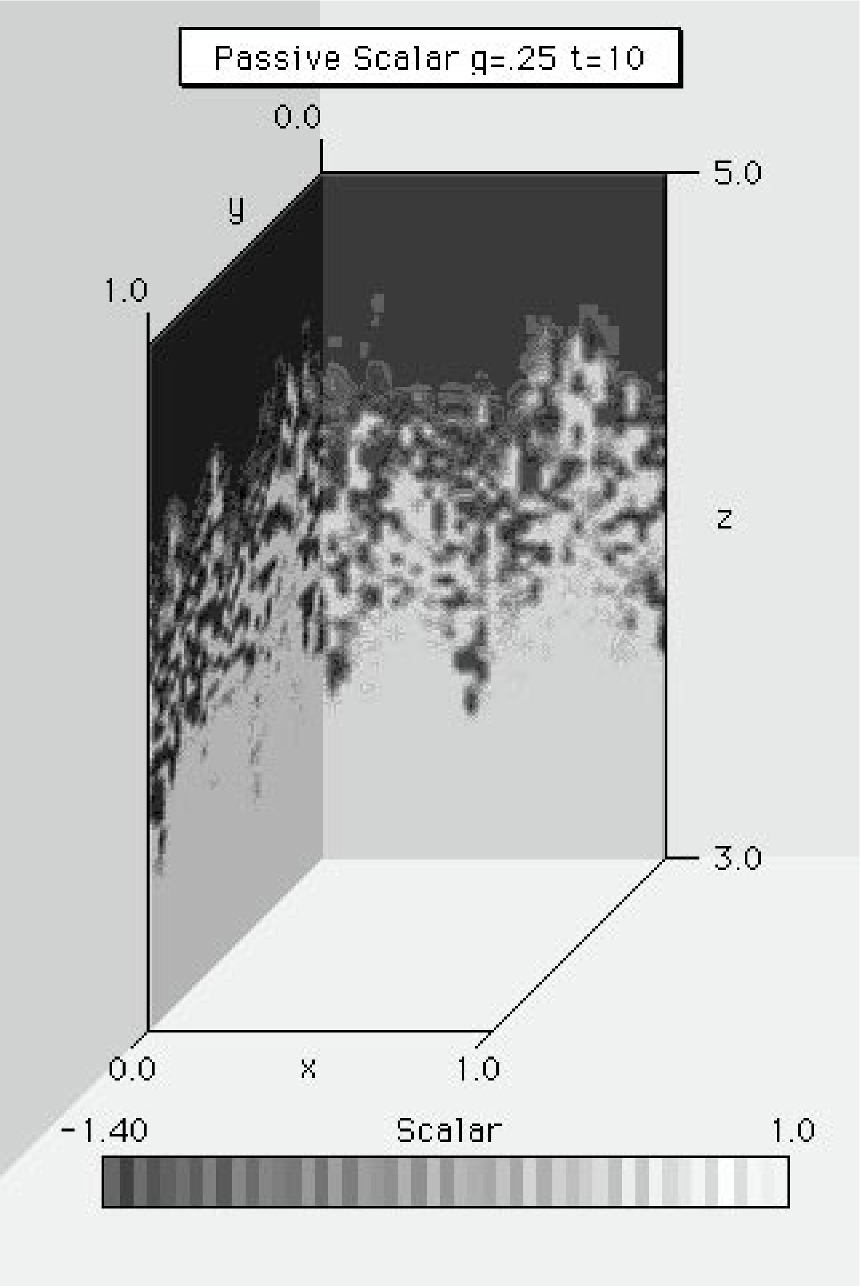


FIGURE 36. Final condition (contour plot)



By t=10, this structure has disappeared. The mixing is almost totally random. Convective motion has mixed the two fluids with the help of the eddy-viscosity model which has handled the subgrid-scale effects.

FIGURE 37. Final condition (3-D view)



These set of runs prove the first three points, for the given range of parameters tested. The stochastic backscatter term has broken the stationarity of the unstably stratified Rayleigh-Taylor problem and induced random mixing. This random mixing has the correct behavior in that it is a function of t^2, and the values of the dimensionless coefficient ξ are in very close agreement with the results of others. Finally, this constant is relatively invariant with respect to the variation of gravity and the smoothing parameter.

14.2 Behavior of the Random Mixing Layer with Respect to C_B

The final test done is a parameter study of the eddy-viscosity model. Various values of the two main constants, C_S and C_B, are used in a computation of the random mixing of the interface. The smoothing parameter is held constant at $\alpha = 25$, and gravity is set at $g = 0.25$; computations are run to time $t_{\max} = 10$.

The purpose of this study is twofold. The first purpose is to see how sensitive the mixing layer evolution is to the value of the backscatter constant. The other purpose is to determine if there is a ratio that the Smagorinsky constant must maintain with respect to the backscatter constant in order to get acceptable computations.

TABLE 15. Parameter check for Smagorinsky and backscatter constants

ξ	$C_S = 0.1$	$C_S = 0.2$	$C_S = 0.4$
$C_B = 0.1$	.0750[a]	.0520	.0250
$C_B = 0.2$	.0613[b]	.0575	.0375
$C_B = 0.4$	------------------->	[c]	.0500[d]

[a] This ran stably to time 6.9

[b] This ran stably to time 4.7

[c] The two runs with large backscatter constant and small Smagorinsky constants quickly went unstable.

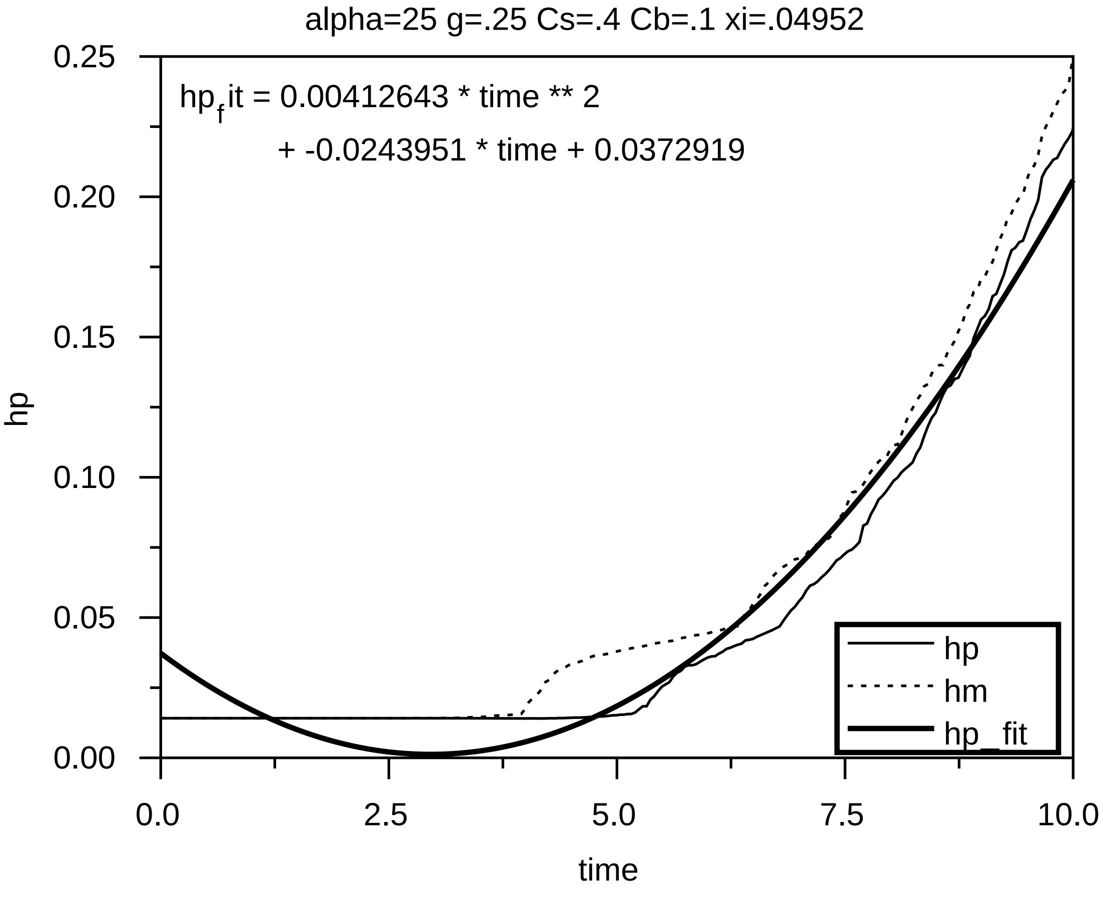
[d] This ran stably to time 8.85

Table 15 lists the results of the test runs, which are in the same form as Table 14 ; the values for ξ are from normalized time plots. Several important observations can be drawn from this data.

The first thing to note is that when the backscatter constant is larger than the Smagorinsky constant, the computation runs into trouble. The growth of the fluctuations associated with the random forcings grow too fast for the eddy-viscosity terms to keep up with. This leads to instabilities which result in arithmetic overflow. The larger C_B is compared to C_S, the faster this occurs. The runs where the ratio is 4:1 and 2:1 where $C_B = 0.4$ went unstable rather quickly (too short to gather any useful data to estimate ξ.) Even in the 1:1 case, the computation only made it to about time 8.9. By these test cases, it can be concluded that $C_B = 0.4$ is just too big; too much random forcing is being induced for the numerics and eddy -viscosity model to handle. Likewise, runs with a ratio of 2:1 with $C_B = 0.2$ could only go to a time of about 4.7.

The next point covers the case where the ratio is 1:1. These runs give the best results. When the Smagorinsky constant is larger that the backscatter constant, the results are also acceptable, but as Figure 38 and Figure 39 show, it takes quite a long time to break the symmetry. There is simply too much damping as compared to forcing at the smaller scales. In the 4:1 case shown, it takes almost to time 5 to see the mixing layer grow larger than the grid size. Compare this to the 1:1 case Figure 30 of and Figure 31, where it takes half as long. Two effect contribute to this: the smaller size of C_B; and the much larger relative size of C_S. Other runs (whose plots are not shown here) also bear this out.

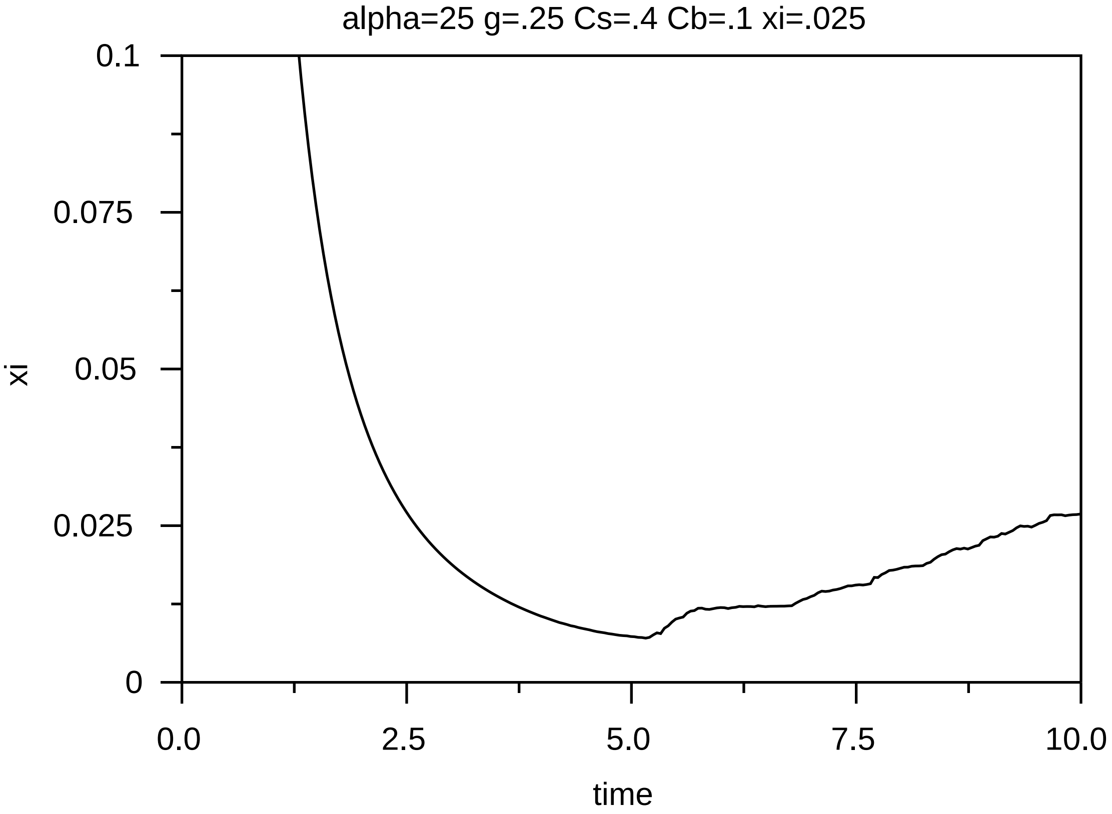
FIGURE 38. Mixing length of spike and bubble for large ratio of constants



In summary, several conclusions can be drawn about the behavior of the eddy-viscosity/ stochastic backscatter model:

1. It is best to keep the ratio of the constants as close to 1:1 as possible, or at least keep the eddy-viscosity the larger of the two.
2. When the stochastic backscatter constant is too large (0.4 in this case), too much random forcing results in numerical overflow.
3. When the eddy-viscosity constant is too small (0.1 in this case), not enough damping leads to the same type of instability.
4. Within the acceptable ranges of C_B and C_S, ξ is fairly invariant; however there does seem to be some sensitivity to the ratios of C_B and C_S.

FIGURE 39. Evolution of ξ for large ratio of constants



15.0 Rayleigh-Taylor Simulation Conclusions

In this chapter, the major goal has been to make a parameter study of the random mixing layer problem for the Rayleigh-Taylor instability. To get there with the code developed in the previous chapter “Programming Models and Timing Studies” , three major steps were taken.

The first step was to develop a method to assist the MacCormack method in handling the substantial density and internal energy discontinuities of the initial conditions. This was achieved in two parts. The first part was by the addition of an artificial viscosity to the numerical method. This was the Smagorinsky eddy-viscosity, which had been modified for the compressible hydrodynamics by the addition of a buoyancy and compressibility terms. The second part was by smoothing the interface in an analytically consistent manner.

Using these two techniques, it was then possible to test the computational code against a known analytic solution. This was the linear perturbation solutions to the Rayleigh-Taylor problem. Without the artificial viscosity, and the smoothing of the interface, the MacCormack scheme performance was very poor. Dispersive errors generated erroneous acoustic pulses which rendered the calculation meaningless. With their addition however, the errors were reduced to a manageable level. This allowed for reasonable computations to be done, as long as the compressibility was kept small enough to prevent the formation of shocks.

Several weaknesses were still present however, suggesting further work. The dispersive errors could not be fully eliminated, suggesting that an addition to the MacCormack method of a Flux Corrected Transport or TVD scheme might be in order. These two monotone methods may reduce or eliminate the remaining dispersive errors. And the full development of the analytical solution of the linear perturbation problem for a smoothed interface would give an exact answer for the growth rate in this case. However, neither of these were necessary to go on to the next phase of the work at hand.

The final portion of this chapter was the actual study of the random mixing layer. The addition of stochastic backscatter to the eddy-viscosity model allowed for two things: the addition of a needed “fix” for the Smagorinsky eddy-viscosity; and the non-intrusive method needed to break the symmetry of the stationary unstable interface and initiate random mixing. This method worked very well; the dimensionless mixing coefficient determined by the simulations here agreed with that found in laboratory experiments by Youngs and Read [87][112][113][114].

This good performance was the result of several interacting factors. The first factor was the fact that the density transition at the interface was smooth. Secondly, the starting transient errors in the velocity field are less important for the overall evolution in this case. In the linear perturbation studies, any error in the velocity calculations contributed to errors in the large scale structure; here the details of the velocities are very important. But in the randomly perturbed interface, there are no overall large scale velocity structures which are important. The velocity field (via the random accelerations of the stochastic terms) was random, so any error introduced by the numerics can be thought of as just another "random" forcing. Thus the method of generating the velocity field "masked" out any errors the numerics may have introduced, resulting in the excellent performance of the code for this problem. The final factor was that the eddy-viscosity model can handle the smaller, sharp fluctuations which generate predominantly shear stresses, better than the acoustic errors (compressive) which were dominant in the linear calculations. An improvement in the compressive portion of the eddy-viscosity model should help with this difficulty. None the less, all these effects contributed to the excellent performance in the random mixing layer problem.

Several important, new results came out of this final portion of the study:

1. The stochastic backscatter term is successfully coupled with the eddy-viscosity model.
2. The stochastic backscatter term properly breaks the symmetry of the stationary unstable interface.
3. The previous estimates of the magnitude of the backscatter constant (0.2) and the Smagorinsky constant (0.2) turn out to be proper for this problem. Within a (small) range the problem's evolution is not directly dependent on the values of these constants.
4. Too much backscatter, or not enough damping, leads to numerical overflow.

While the fourth point was previously suspected and suggested by the work of others, the first three items are quite interesting and new. Of special interest is the fact that in this problem, stochastic backscatter was able to do two jobs for the price of one: it is a proper and necessary addition to the subgrid scale model; and it can mimic the microscopic fluctuations which initiate random mixing.

Conclusions

16.0 Final Results of this Research

There were three major components to the research performed for this dissertation.The first component was the development of the physical/computational model. The Navier-Stokes equations where chosen as the basis owing to the inclusion of viscous terms since it was desired to include an eddy-viscosity model. The numerical scheme chosen was based on Lax-Wendroff methods. There is a lot of research on this family of numerical PDE solvers, including a substantial effort (not reported here) by the author to analyze the behavior of the method for non-linear, conservative forms. This resulted in a familiarity with the methods that proved valuable in both coding, and analyzing the results. And because of their simplicity, it was felt that they would be particularly suited to implementation in data parallel and domain decomposition computer codes on parallel computers.

After the basic serial and vectorizable forms of the code were developed, the question arose: what is the best programming paradigm for putting this on a state of the art supercomputer? Since the author had access to the Los Alamos CM-5, a study of the various programming models on this machine was initiated. The CM-5 supports both message passing and data parallel approaches. It was found, however, even after much fine tuning and correspondence with Thinking Machines Inc., that the data parallel approach was substantially better (in terms of speed, memory usage, and ease of programming/debugging.) Thus the final version of the computer program became the data parallel code. However, while this is the best choice for the CM-5 given the chosen problem, it may not be for another problem or machine.

The final portion of this dissertation was a study of the Rayleigh-Taylor problem. This problem pushed the capabilities of the Lax-Wendroff methods to the limit. In a range of small Atwood numbers and low compressibility, the MacCormack scheme was able to handle the computation well. But for long times, and for large density discontinuities, the calculations were unstable. Acoustic pulses generated at the interface, combined with discretization errors of the boundary and initial conditions so on overwhelmed the computation. The addition of the eddy-viscosity, and the smoothing of the density transition at the interface were two methods used to push the envelope of the allowable parameters which the Lax-Wendroff methods could handle. The results were acceptable, but still disappointing. The eddy-viscosity terms allowed computations to go on for much longer periods of time: up to the limits imposed by having fixed boundaries which the bubbles and spikes would hit, or extreme roll-up of the mushrooming bubbles. The smoothing of the interface did permit larger jumps in density, and if smooth enough totally reduced the acoustic waves to a scale irrelevant to the calculation.

However, the introduction of the smoothing created another problem that has not yet been solved. If the interface is smooth enough to totally eliminate the acoustic pulse, then the growth rate calculated by the analytic sharp interface assumption is no longer valid. This puts the initial conditions generated into criticism. But if the interface is sharp enough that the analytic initial conditions are valid, then the acoustic pulse reappears which invalidates the calculation. This problem will remain for the linear analysis and validation until the analytic (or numerical) solution of the initial conditions can be solved for the much more complicated case of the smooth interface.

The random mixing layer problem was a good test for the addition of stochastic backscatter to the eddy-viscosity model. This addition served two purposes. The first is a necessary addition of a term which can allow the subgrid scales (unresolved) to affect the resolved flow. In the random mixing problem, this results in the second beneficial effect: the random forcings break the symmetry of the initial unstable stationary state and initiate mixing.

The results achieved with this problem were quite good. The traditional value of the dimensionless mixing coefficient was obtained by the code for a reasonable range of values of gravity and smoothing parameters. This value was invariant with these changes, which is in good agreement with experimental results, and other computational codes (which by the way, use substantially more complicated numerical schemes.) The value of this mixing coefficient was also shown to be somewhat invariant to the values of the eddy-viscosity constants. However, it was also shown that these constants are restricted in range; the backscatter constant should be kept smaller than 0.4 to prevent overflow due to excessive forcing and larger than 0.1 to initiate the mixing in a “reasonable” amount of time. The Smagorinsky constants needs to be larger than 0.1 to ensure enough damping of oscillatory modes, but not so large as compared to the backscatter constant to dampen the random fluctuations too much. Good values for both are $C_S = 0.2$ and $C_B = 0.2$, given the parameters and physical conditions used here.

16.1 Needed Improvements and Future Work

In the end, this dissertation had its victories and its disappointments. The timing studies section went rather smoothly. The Rayleigh-Taylor portion was much more difficult. There is quite a bit of room for additional work and improvements. One of the things that is evident in completing this work is the amount of additional work that could or should be done!

In the timing section, one major additional piece of work comes to mind for this author. With the aid of the compiler, it is possible to disassemble the compiled code. This could be done on the CMF/CMMD program in order to see how the *cmf$ align* and *cmf$ layout* directives were actually placing the arrays in the message passing subroutines. Further dissection in this manner might lead to other insights as to why the timings of the gather/scatter routines varied so greatly. It should be useful to the writers of the CMF complier and CMMD libraries to understand why this bottleneck appears in the message passing codes.

The work done in with the Rayleigh-Taylor instability really opened up many vistas for extra work. The physical model used could be improved by the addition of a surface tension term. This surface tension might help to control some of the surface wave (acoustic waves) which occur because of the interface, and could help to dampen some of the sharp growths there. Some of the previously cited works include surface effects such as this, others ignore it. While it was a simplification to ignore it in this work, the surface tension is an important physical effect at the interface between two fluids.

The numerical scheme should be improved to handle the discontinuities better. This would lead to a better resolution of the linear growth rate, and over all it should fix the dispersive errors that cause the time scale problems. Several methods come to mind as fixes. The two that should be tried first are the FCT[1] and the TVD[2] schemes as an add-on to the MacCormack scheme. This entails quite a bit of work however, since most of the published literature for these methods deals mainly with linear advection-diffusion problems. Extensions do exist for the Euler equations, but this author is unaware of extensions to the Navier-Stokes equations. If the MacCormack scheme still proves to be unsatisfactory with these additions, the next step would be to try the Godunov schemes.

The problem that remains with the divergence term in the extension of the Smagorinsky eddy-viscosity model still needs to be sorted out. While previously it has been set to zero, in this research it could be used somewhat successfully at magnitudes of approximately unity. In this capacity it did help control undesired compressive acoustic waves, but it was very touchy and likely to cause instability. This needs to be fixed, since a compressive term for the compressible form of the eddy-viscosity model is a necessity.

Finally, more testing needs to be done with the stochastic backscatter model. While it was used successfully in the random mixing problem (and other work by other authors,) the model used here is still in its infancy. Direct numerical simulations need to be run with real world problems, and comparisons made with the large-eddy simulations of the same flows. Some of this work has been done for isotropic, homogeneous turbulence, but a lot more remains to be done.

As a final closing word, one of the interesting things that this dissertation research has taught me is this:

> The more I seem to accomplish, the more there is that needs to be done!

[1] Flux Corrected Transport

[2] Total Variation Diminishing

Appendix A - The Equations of Fluid Flow

17.0 The Navier-Stokes Equations

The Navier-Stokes equations are five non-linear, coupled, second order partial differential equations which are used to model three-dimensional fluid flows within certain restrictions. There are first-order temporal derivatives, and first- and second-order spatial derivatives. The equations exhibit the properties of parabolic, elliptic, and hyperbolic partial differential equations depending on the flow regime. There are analytic solutions for some specific cases, but in general the analytic solutions are not know. In fact, it has not yet been proved whether the set of equations are even well-posed[1]! This of course has serious implications (practical and philosophical) for attempts at solutions to these equations.

There is a plethora of literature concerning the Navier-Stokes equations. Texts seem to be of three general varieties: undergraduate engineering texts [106]; graduate level engineering and physics texts [52][53][80][93]; and mathematical investigations of the properties of the equations[22][74].

Below is a summary of the Navier-Stokes equations, and the set of constitutive relationships used in this research to close the set of equations. This is then followed by a discussion of the Euler equations, which are often used to model the Navier-Stokes equations when the fluid considered has negligibly small molecular viscosity.

17.1 Variables

There are many variables used to characterize three-dimensional fluid flow. Since there are five equations, there must be five independent variables. These are often chosen to be the three velocity components, density, and pressure. Since the last two define thermodynamic properties, they can then be used to find any other desired thermodynamic quantities (i.e., internal energy, temperature, entropy, etc.)

[1] This a necessary condition for uniqueness and existence theorems.

In the following, the summation convention is assumed. This means that repeated indices in a term implies summation from one to three, i.e. $u_k u_k = u_1 u_1 + u_2 u_2 + u_3 u_3$. Also, the positional vector $\vec{x}$ is the same as (x, y, z) or (x_1, x_2, x_3). This is analogous to the tensor notation x_i. Likewise, the velocity vector $\vec{u}$ is the same as (u, v, w), or (u_1, u_2, u_3), or the tensor notation u_i.

TABLE 16. Common Variables

$u_i(\vec{x}, t)$	velocity in the i direction
$\rho(\vec{x}, t)$	density
$P(\vec{x}, t) = P(\hat{u}, \rho)$	thermodynamic pressure
$\hat{u}(\vec{x}, t)$	internal energy
$E(\vec{x}, t) = \rho\left(\hat{u} + \frac{1}{2}u_k u_k\right) = \rho e$	total energy
$\tau_{ij}(\vec{x}, t)$	stress tensor
$q_i(\vec{x}, t)$	heat flux
$f_i(\vec{x}, t)$	external force

17.2 Constants

The following quantities are considered constant for the purpose of this research. Many of these “constants” are in fact not constant if pressure, temperature, etc. is allowed to vary sufficiently. However, the flows studied here are not within these extreme regimes.

TABLE 17. Common Constants

γ	specific heat ratio
μ	molecular viscosity
μ_v	bulk viscosity
λ	second coefficient of viscosity
α	thermal diffusivity

17.3 Dimensionless Numbers

There are several important dimensionless numbers in the study of fluid flow. The first is the Reynold's number Re, which is a ratio of convective to viscous momentum transport affects:

$$Re = \frac{ul}{\nu} \qquad \text{(EQ 15)}$$

u is a characteristic velocity scale and l is a characteristic length scale. $\nu = \frac{\mu}{\rho}$ is the kinematic viscosity coefficient. When Re is large, the convective effects are far more important than the viscous effects in the transport of momentum. These flows are called *turbulent*. At the other extreme is the *laminar* flows; these flows tend to be of very viscous fluids.

The Prandtl number Pr is another useful dimensionless number; it is the ratio of viscous to thermal diffusion:

$$Pr = \frac{\nu}{\alpha} = \frac{\mu C_P}{\kappa} \qquad \text{(EQ 16)}$$

where C_P is the constant pressure specific heat capacity, and κ is the thermal conductivity coefficient. For an ideal gas, kinetic theory predicts a Prandtl number of $\frac{2}{3}$, and this is closely confirmed by experiment.

The Schmidt number is a ratio the diffusive transport of momentum to that of the mass of a species. It is especially useful in buoyancy driven flows and reactive flows where there are more than one species of fluid or gas. It is defined by:

$$Sc = \frac{\nu}{D} \qquad \text{(EQ 17)}$$

where D is the molecular diffusivity coefficient.

The Lewis number is the ratio of the Schmidt number to the Prandtl number. It is therefore the ratio of heat transport to mass transport. Mathematically,

$$Le = \frac{\alpha}{D} \qquad \text{(EQ 18)}$$

In many systems, the Lewis number Le is very close to one. This implies that the Schmidt number equals (approximately) the Prandtl number.

The Mach number Ma is the ratio of the local velocity to the local sound speed. Since the sound speed $c^2 = \left.\frac{\partial P}{\partial \rho}\right|_S$ is a thermodynamic variable, this makes Ma a thermodynamic quantity too. For an ideal gas, the sound speed c is given by:

$$c^2 = \gamma\frac{\Re T}{M} = \gamma\frac{P}{\rho} \qquad \text{(EQ 19)}$$

where $\Re$ is the universal gas constant, M is the molecular weight of the gas, and T is temperature. The final definition for Mach number then becomes

$$Ma = \frac{|u|}{c} \qquad \text{(EQ 20)}$$

where the sound speed c is defined above and $|u|$ is the local speed of the flow. Flows at speeds below the sound speed c have $Ma < 1$ and are called subsonic. Flows with $Ma > 1$ are called supersonic. It is typically held that a compressible flow with $Ma < 0.4$ is effectively incompressible and will not form any type of shock.

Besides characterizing the flow, these dimensionless numbers serve another useful function. It is typically possible to cast the equations of motion into a non-dimensional form. All of the variables and constants then reduce to these aforementioned dimensionless numbers (plus possibly a few more.) Where originally there may have been a dozen adjustable parameters in the dimensional form of the equations, there are now just a few of these dimensionless numbers. This greatly simplifies parameter studies.

17.4 Conservation Form of the Equations

17.4.1 Conservation of Mass

The following equations are the compressible Navier-Stokes equations. It is assumed that there are no external heat sources, or radiative heat transfer. Only one species is involved, so there are no chemical source or reaction terms, and only a single continuity equation is necessary. The fluid is assumed to have non-zero (although perhaps very small) molecular viscosity, and obeys the Ideal Gas Law relation. The fluid is assumed to be Newtonian, with the corresponding stress tensor constitutive relationship. The fluid is also assumed to obey the Fourier Law of heat conduction.

With these assumptions, the fluid motion is assumed to be exactly described by five partial differential equations. The first equation is *the Continuity Equation*, based on the physical principle of conservation of mass. Since there is only one species with no chemical reactions, no species conservation equations are needed. This equation states that in a control volume, no matter is created or destroyed; any change in mass inside the control volume must be the result of convection across the volume's boundary.

$$\frac{\partial\rho}{\partial t} + \frac{\partial}{\partial x_k}(\rho u_k) = 0 \tag{EQ 21}$$

17.4.2 Conservation of Momentum

The next three equations are for the three velocity components, commonly called *the Momentum Equations* or *the Equations of Motion*. They are based on the principle of conservation of momentum. Basically, they are just acceleration equations for a control volume. Momentum can be carried across the volume's boundary by convection. Momentum can be removed by viscous dissipation. There is also a buoyancy acceleration resulting from external forces and density gradients. In incompressible flows it is necessary to model this effect with a Boussinesq type approximation, but since the compressible equations explicitly include density variations, such an approximation is not necessary here.

$$\frac{\partial}{\partial t}\rho u_i + \frac{\partial}{\partial x_j}(\rho u_i u_j - \tau_{ij}) = \rho f_i \tag{EQ 22}$$

17.4.3 Conservation of Energy

The final equation is *the Energy Equation*. Essentially, it is the First Law of Thermodynamics, in a control volume format. Three types of “actions” can occur in this control volume. Energy can be advected across the volume's boundary, or moved about within. In viscous fluids, kinetic energy (motion) is converted to heat (internal energy) due to the action of viscous dissipation. Finally, external forcing can add energy to the system. The final equation for total energy per unit volume is:

$$\frac{\partial}{\partial t}\rho e + \frac{\partial}{\partial x_k}(\rho e u_k - \tau_{ik} u_i + q_k) = \rho u_k f_k \tag{EQ 23}$$

18.0 Constitutive Relationships

The previous five equations define the motion of a one-component fluid. Unfortunately, the set of equations is not closed. There are 15 unknown quantities (density, three velocity components, total energy, pressure, three heat flux terms, and six independent stress tensor terms,) leaving a deficit of ten equations to close the system.

The constitutive relationships fill this gap. These consist of six independent equations for the stress tensor, three equations for the heat flux, and one equation of state.

18.1 Newtonian Stress Tensor

The stress tensor is determined by the Newtonian fluid assumption:

$$\tau_{ij} = \left(-P + (\lambda + \mu_v) \frac{\partial u_k}{\partial x_k} \right) \delta_{ij} + \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \qquad \text{(EQ 24)}$$

For an isotropic Newtonian fluid, the viscosities are related by $\mu_v = \lambda + \frac{2}{3}\mu$. The bulk viscosity μ_v is assumed to be zero by the Stokes hypothesis. The final simplified form of the stress tensor then is:

$$\tau_{ij} = \left(-P - \frac{2}{3}\mu \frac{\partial u_k}{\partial x_k} \right) \delta_{ij} + \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \qquad \text{(EQ 25)}$$

Notice that the first term contains the divergence of the velocity field; this is zero for incompressible fluids.

18.2 Fourier’s Law

The heat flux term is determined by the Fourier law assumption:

$$q_i = -\kappa \frac{\partial T}{\partial x_i} \qquad \text{(EQ 26)}$$

where T is absolute temperature, and κ is thermal conductivity. This form also assumes no thermal radiation, and no thermal source terms.

18.3 Equation of State

The equation of state is a relationship which gives the functional dependence of a thermodynamic variable upon two other independent thermodynamic variables. Usually it is thought of as relating density (or specific volume) and temperature to pressure. A commonly used equation of state is that of an ideal gas:

$$PV = RT \tag{EQ 27}$$

or

$$P = \rho RT \tag{EQ 28}$$

In these equations P is pressure, V is specific volume, $\rho = \frac{1}{V}$ is density, R is the gas constant, and T is the absolute temperature.

An ideal gas is a dilute gas made of monatomic particles (for which $\gamma = \frac{5}{3}$.) We further assume that we deal with a perfect gas, i.e. that R is a constant. Note that R is equal to the universal gas constant $\Re$ divided by the molecular weight M of the gas.

18.3.1 Derivation of Internal Energy Relationships

Internal energy $\hat{u}$ is a thermodynamic variable and thus can be related to two other thermodynamic variables. Assume that

$$\hat{u} = \hat{u}(P, T) \tag{EQ 29}$$

Then the thermodynamic differential of internal energy is

$$d\hat{u} = \left.\frac{\partial}{\partial P}\hat{u}\right|_T dP + \left.\frac{\partial}{\partial T}\hat{u}\right|_P dT \tag{EQ 30}$$

The first partial derivative on the right hand side will now be investigated in detail. Start with the Gibbs' function,

$$d\hat{u} = Tds - PdV \tag{EQ 31}$$

where s is the entropy. Taking the partial derivative with respect to P while holding T constant gives:

$$\left.\frac{\partial}{\partial P}\hat{u}\right|_T = T\left.\frac{\partial s}{\partial P}\right|_T - P\left.\frac{\partial V}{\partial P}\right|_T \qquad \text{(EQ 32)}$$

Now using the Maxwell relation

$$\left.\frac{\partial s}{\partial P}\right|_T = -\left.\frac{\partial V}{\partial T}\right|_P \qquad \text{(EQ 33)}$$

gives finally

$$\left.\frac{\partial}{\partial P}\hat{u}\right|_T = -T\left.\frac{\partial V}{\partial T}\right|_P - P\left.\frac{\partial V}{\partial P}\right|_T \qquad \text{(EQ 34)}$$

Now assuming the gas in question is ideal and perfect (see Equation 27) it is evident that V is only a function of the ratio of T and P. Thus evaluating Equation 34 gives

$$\left.\frac{\partial}{\partial P}\hat{u}\right|_T = \frac{-TR}{P} + \frac{RT}{P} = 0 \qquad \text{(EQ 35)}$$

Thus Equation 30 evaluates to

$$d\hat{u} = \left.\frac{\partial}{\partial T}\hat{u}\right|_P dT \qquad \text{(EQ 36)}$$

or more simply; $\hat{u}$ is only a function of T.

18.3.2 Relationship to Gas Kinetic Quantities

For the perfect ideal gas the gas constant is $R = c_P - c_V$ where c_p is the specific heat at constant pressure and c_V is the specific heat at constant volume. The ratio of the specific heats is

$$\frac{c_p}{c_V} = \gamma \qquad \text{(EQ 37)}$$

which is constant under the perfect gas assumption.

Now the monatomic gas only has three degrees of freedom (motion in the x, y, and z directions.) There are no spin states or internal vibrational degrees of freedom since each particle consists of only one atom. This gives a molecular energy of

$$\bar{\varepsilon} = 3\frac{1}{2}kT = \frac{3}{2}kT \qquad \text{(EQ 38)}$$

where k is the Boltzmann constant

$$k = \frac{\Re}{N_a} \qquad \text{(EQ 39)}$$

and N_a is Avogadro’s number. Thus the total internal energy (per unit mass) is

$$\frac{N_a}{M}\frac{3}{2}\frac{\Re}{N_a}T = \frac{3}{2}\frac{\Re}{M}T = \frac{3}{2}RT = \frac{3}{2}PV = \frac{3}{2}\frac{P}{\rho} = \hat{u} \qquad \text{(EQ 40)}$$

This allows the evaluation of the specific heat ratio for a perfect ideal gas as $\gamma = \frac{5}{3}$.

18.3.3 Some Useful Relationships

The equation of state used for pressure thus comes from Equation 40

$$P = (\gamma - 1)\rho\hat{u} \qquad \text{(EQ 41)}$$

A second useful relationship converts the normal definition of heat flux

$$q_i = -\kappa\frac{\partial T}{\partial x_i} \qquad \text{(EQ 42)}$$

into an equation dealing with internal energy. First note that from Equation 40 that

$$T = \frac{P}{R\rho} \qquad \text{(EQ 43)}$$

The definition of the dimensionless Prandtl number is

$$Pr = \frac{\nu}{\alpha} = \frac{\mu c_p}{\kappa} \qquad \text{(EQ 44)}$$

where ν is the kinematic viscosity, α is the thermal diffusivity, and μ is the molecular viscosity.[2] Substituting Equation 44 and Equation 43 into Equation 42 gives the new form of the heat flux as:

$$q_i = -\frac{\mu c_p}{Pr}\frac{(\gamma - 1)}{R}\frac{\partial}{\partial x_i}\left(\frac{P}{(\gamma - 1)\rho}\right) = -\frac{\mu\gamma}{Pr}\frac{\partial}{\partial x_i}\hat{u} \qquad \text{(EQ 45)}$$

A third useful relationship will not be derived here, but the result will be simply stated. For a perfect ideal gas the Prandtl number can be derived exactly from kinetic theory, as

$$Pr = \frac{2}{3} \qquad \text{(EQ 46)}$$

Finally, the relationship between internal and total energy will be given. If total energy per unit mass is e, then

$$\rho e = \rho\hat{u} + \frac{1}{2}(\rho u^2 + \rho v^2 + \rho w^2) \qquad \text{(EQ 47)}$$

where u, v, and w are the x, y, and z components of velocity respectively.

19.0 The Euler Equations

The Euler equations can be derived in a number of independent ways. However, the easiest way to think of the Euler equations is as describing a gas flow with zero viscosity and zero thermal conductivity. At a molecular level this implies that there is no collision mechanism whereby two particles may interchange momentum or heat through a dissipative mechanism. In fact, the only remaining method of transferring energy from kinetic to internal is through pressure, i.e. through a PdV term. Of course, energy may still be advected by the velocity field.

The Euler equations are thus a model for fluid flow in which the molecular interactions occur on a time scale quite large compared to that of the motion of the fluid, or at a very

[2] For an incompressible fluid $c_p = c_V$. This replacement can lead to confusion.

low density. Thus the Euler equations are often used as a model of gas dynamics, especially shock dynamics. However, several points must be brought forth.

The first is quite significant. Since the Euler equations have no dissipative mechanism, can they exhibit turbulent behavior? This is a very controversial point which has believers of an almost religious fervor on both sides of the issue! Some believe that since the viscosity is set to zero in the Navier-Stokes equations to derive the Euler equations, this is the same as saying the Euler equations are a flow with infinite Reynold's number - a very turbulent regime indeed. However, others point out that the behavior of the Navier-Stokes equations in the limit of viscosity going to zero can be phenomenologically different than the behavior of the Euler Equations (where the viscosity has been set to zero.) For instance, since the Euler equations have no dissipative mechanism it is impossible to see kinetic energy converted to internal energy by shear stresses - something which can occur in turbulent flows using the Navier-Stokes equations. However, the two sets of equations can still exhibit similar behavior. At high Mach numbers both sets of equations give rise to shocks. The point is that one must be careful how one interprets the behavior of the Euler equations. In this dissertation, the issue is not directly addressed. The Navier-Stokes equations are used, simply because they are the basis for adding the eddy-viscosity terms needed in a large-eddy simulation.

The second major point is a reality check. No flow in the real world can have zero viscosity or zero thermal conductivity; it is physically impossible. The fundamental laws of physics guarantee that if two particles of different momentum collide, there will be some type of interaction. The same holds for particles at different temperature. Even in a rarefied gas, there are still millions of collisions per second. For the Navier-Stokes equations to be applicable (and similarly for the Euler equations) a continuum assumption must hold. If the gas is so rarefied that collisions are not occurring frequently, it is hard to justify this continuum assumption. One should then be using the Boltzmann and kinetics equations instead of continuous methods.

Still, when all is said and done, the Euler equations are often used to model gas flows. As long as the flow regime has a large Reynold’s number so that convective effects far exceed dissipative effects, the Euler equations are *held* to be a good model for the Navier-Stokes equations. Thus the Euler equations are often used in modeling turbulent flows. In this case, the artificial viscosities added for the turbulence model are orders of magnitude larger than the molecular viscosities/diffusivities in the Navier-Stokes equations. This renders the effects of the physical dissipative mechanisms irrelevant when compared to the modelled effects, thus they are then ignored.

One of the most interesting things about the Euler equations is the fact that there are so many ways to derive them. The Euler equations can be derived from macroscopic gas dynamics [51][101][107], from the Boltzmann Equation assuming no collision operator [104][107], or from the Navier-Stokes Equations by setting viscosity and thermal conductivity to zero [53]. They can also be derived from variational principles [115], from the view point of differential forms [30], or from continuum mechanics [94].

19.1 Conservation Form of the Equations

The Euler equations can easily be derived from the Navier-Stokes equations by setting the viscosity and thermal diffusivity coefficients to zero. The resultant equations are listed below. The only other equation needed to close the system is an equation of state.

19.1.1 Conservation of Mass

$$\frac{\partial\rho}{\partial t}+\frac{\partial}{\partial x_k}\rho u_k = 0 \qquad \text{(EQ 48)}$$

19.1.2 Conservation of Momentum

$$\frac{\partial}{\partial t}\rho u_i+\frac{\partial}{\partial x_j}(\rho u_i u_j+P\delta_{ij}) = \rho f_i \qquad \text{(EQ 49)}$$

19.1.3 Conservation of Energy

$$\frac{\partial}{\partial t}\rho e+\frac{\partial}{\partial x_k}(\rho e u_k+P u_i\delta_{ik}) = \rho u_k f_k \qquad \text{(EQ 50)}$$

19.2 Euler Equations in One-Dimension

In one dimension these constitute three equations from gas dynamics representing conservation of mass, momentum, and energy. There is no viscosity (momentum diffusion) or thermal conductivity (heat diffusion) or external forcing. The most common way of writing these equations is:

$$\frac{\partial}{\partial t}\begin{bmatrix}\rho\\ m\\ E\end{bmatrix}+\frac{\partial}{\partial x}\begin{bmatrix}m\\ \dfrac{m^2}{\rho}+P\\ (E+P)\dfrac{m}{\rho}\end{bmatrix}=0 \qquad \text{(EQ 51)}$$

Here, ρ is density, $m = \rho u$ is momentum with velocity u, E is total energy (internal plus kinetic energy per unit volume), and P is pressure. Since these are three equations with four unknowns, the system is not closed. An equation of state is needed in order to relate pressure to the other quantities; here the ideal gas relation

$$P = (\gamma-1)\left(E-\frac{1}{2}\frac{m^2}{\rho}\right) \qquad \text{(EQ 52)}$$

is used. The constants γ is the specific heat ratio, which for a monatomic gas is $\frac{5}{3}$ and for air is commonly taken as 1.4.

19.3 Euler Equations in Two-Dimensions

In two dimensions, a fourth quantity $n = \rho v$ is added where v is the velocity in the y direction. The Euler Equations then look like:

$$\frac{\partial}{\partial t}\begin{bmatrix}\rho\\ m\\ n\\ E\end{bmatrix}+\frac{\partial}{\partial x}\begin{bmatrix}m\\ \dfrac{m^2}{\rho}+P\\ \dfrac{mn}{\rho}\\ (E+P)\dfrac{m}{\rho}\end{bmatrix}+\frac{\partial}{\partial y}\begin{bmatrix}n\\ \dfrac{mn}{\rho}\\ \dfrac{n^2}{\rho}+P\\ (E+P)\dfrac{n}{\rho}\end{bmatrix}=0 \qquad \text{(EQ 53)}$$

The equation of state is slightly modified too:

$$P = (\gamma - 1)\left(E - \frac{1}{2}\left(\frac{m^2}{\rho} + \frac{n^2}{\rho}\right)\right) \tag{EQ 54}$$

The extensions to three dimensions are reasonably obvious.

20.0 Passive Scalars

Passive scalars satisfy the linear advection-diffusion equation. Besides being used as a simple model of the Navier-Stokes equations, passive scalars are often added to the solution of fluid flow problems as a diagnostic tool.

For instance, in the Rayleigh-Taylor problem there are two fluids separated by an interface. It is possible to define a passive scalar that is equal to +1 on one side of the interface, and -1 on the other. The velocity field advects (moves) this scalar around as the flow evolves. Thus it is possible to follow the new positions of the interface, and to observe any mixing which is occurring. In this case the passive scalar can be thought of as a marker: color for instance. One fluid is red (+1) and the other is blue (-1.)

The evolution equation (conservative form) satisfied by the passive scalar is:

$$\frac{\partial \Phi}{\partial t} + \frac{\partial}{\partial x_i}(u_i \Phi) = \frac{\partial}{\partial x_i}\left(\kappa_\phi \frac{\partial \Phi}{\partial x_i}\right) \tag{EQ 55}$$

where κ_ϕ is a diffusivity coefficient for the passive scalar, and Φ is the scalar itself. κ_ϕ can be set to zero if it is only desired to advect the scalar without any diffusion. However, it may be set to a small value to help remove oscillations which appear due to numerical truncation errors in the numerical scheme.

Equation 55 is the conservative evolution equation for a passive scalar. A more common form of the evolution equation for a passive scalar is for an incompressible flow:

$$\frac{\partial \Phi}{\partial t} + u_i \frac{\partial \Phi}{\partial x_i} = \frac{\partial}{\partial x_i}\left(\kappa_\phi \frac{\partial \Phi}{\partial x_i}\right) \tag{EQ 56}$$

Equation 55 is more general in that the divergence of the velocity field is not assumed to be zero. The second term of the left hand side of Equation 55 expands to:

$$\frac{\partial}{\partial x_i}(u_i\Phi) = u_i\frac{\partial\Phi}{\partial x_i} + \Phi\frac{\partial u_i}{\partial x_i} \qquad \text{(EQ 57)}$$

For an incompressible flow, the second term on the right hand side of Equation 57 equals zero (i.e., the divergence of velocity is zero;) thus Equation 55 becomes Equation 56.

Conceptually, there is one major difference between Equation 55 and Equation 56. In the conservative form, it is possible for the passive scalar to build up; this is a consequence of the non-divergence free velocity field which can now be compressive. In other words, if the scalar is color, it is possible for a point in space to get “bluer” or “redder.” Overall though, the amount of color is conserved. If one area gets “bluer,” another area gets “less blue.” This is not possible in the divergence free form of Equation 56. In this case the scalar is simply moved around by the velocity field; it is impossible for the scalar to locally build up.

The type of passive scalar used in this dissertation is that of Equation 55. The numerics are set up to handle the conservative form of the evolution equations. For consistency, the passive scalar is thus modelled in the same form.

Appendix B - Turbulence Modelling

21.0 Introduction to Turbulence

In many flows of physical interest, there is a wide range of length scales. These length scales range from the size of the physical boundaries, down to the *Kolmogorov length scale* (the length scale at which kinetic energy dissipates to heat.) The motion of the atmosphere has length scales which range from 1000’s of km to less than 1 mm. In three dimensions this is about 10^{20} degrees of freedom. Even a wind tunnel with scales of 10 cm to about 0.1 mm has over 10^{8} degrees of freedom. While some computational problems have been done close to this limit, it is only under conditions of geometric symmetries and low Reynolds number[1] where this is possible. So the question naturally arises: how does one compute a problem with high Reynolds number and thus a wide range of length scales?

The resultant turbulent flow must be modelled. Currently this is a very inexact science; there are few, if any, “good” methods to model turbulent flow. The methods basically break down into one of four general techniques:

1. Statistical models of turbulence
2. Direct numerical simulations
3. Subgrid-scale models
4. Rules of thumb

There is, of course, often overlap between these methods.

Statistical models generally consider a mean flow which is time independent, to which is added a fluctuating flow which contains the turbulent motion. Thus in general the flow must be stationary (or uniform mean flow,) with only a fluctuating sub-component. Often the further restriction of homogeneity and isotropy are imposed to allow simplifications. There is a significant amount of mathematical analysis to back up many of the models;

[1] See Equation 15 in “Appendix A - The Equations of Fluid Flow” on page 83 for the definition of the Reynolds number.

however, in practice the computational results are often still disappointing. Numerous good reference texts detail the work in statistical theories, including those of Lesieur [61], Stanisic [98], Orszag [79], Leslie [62], and the substantial work of Monin and Yaglom [75][76].

Rules of thumb are typically used by engineers. These are useful since they are simple, and thus computationally inexpensive and easy to implement. They are based on curve fitting functions to known velocity profiles, etc. Thus they are mostly applicable in situations that closely match the experiment from which the data were taken; they are severely questionable in cases that deviate from the experimental situation. Schlichting [93] has compiled quite a bit of theory and rules of thumb of this sort in his classic text on boundary layer theory. Another text, Bradshaw, Cebeci and Whitelaw [19], while not solely restricted to rules of thumb, has lots of practical recommendations is.

This dissertation is concerned with the third method (and to a lesser extent, the second.) A direct numerical simulation (DNS) explicitly computes all length scales; it solves *directly* the Navier-Stokes equations. The size of the computational grid thus must be fine enough to capture all significant motion down to the Kolmogorov length scales; the computational domain must be large enough to include all boundaries. As mentioned in the first paragraph, this limits a three-dimensional flow to relatively low Reynolds numbers of order hundreds (given current computer technology.) The following section discusses in depth the third method: subgrid-scale modelling and large-eddy simulations.

22.0 Large-eddy Simulations

A large-eddy simulation (LES) resolves explicitly the larger scales, and treats the influence of all scales below some certain size with a subgrid-scale model. Typically, this “cut-off” length scale is far above the scale of dissipation. Thus the subgrid-scale model must incorporate any turbulent dissipative, diffusive, and reactive processes that occur below the grid size.

There are three broad approaches to large-eddy simulations:

1. spectral methods
2. finite difference methods
3. vortex-dynamics methods

This dissertation will concern itself solely with the finite difference based methods. They use the native Navier-Stokes equations[2] with new viscosity/diffusivity terms in addition to the molecular counterparts; typically these eddy-viscosity terms are orders of magnitude larger than the molecular counterparts. The resultant equations are then solved as in the case of the DNS.

One way to think about this eddy-viscosity is a follows. If molecular viscosity acts at a length scale of 10^{-5} m, and the smallest grid size is of the order 10^{-1} m, is there a way to introduce a new viscosity to act at this larger scale?

FIGURE 40. Kolmogorov Energy Dissipation Spectrum

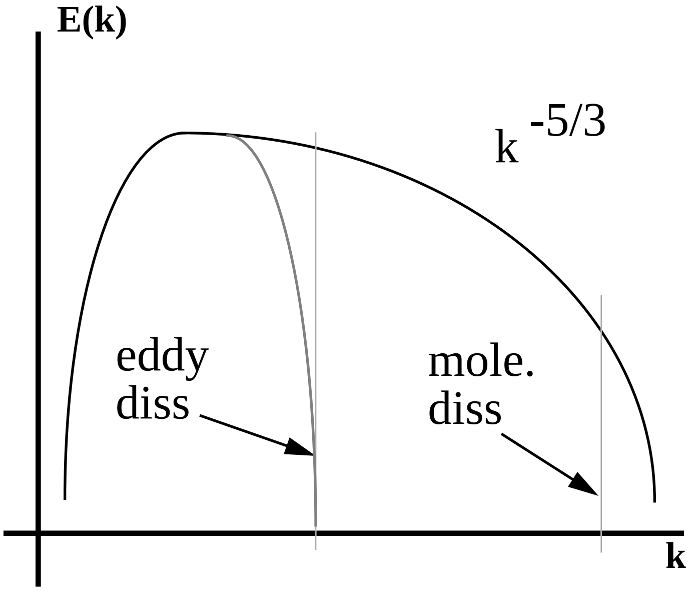


Figure 40 illustrates this concept. This is a graph of kinetic energy per unit wavenumber as a function of wavenumber. The molecular dissipation occurs at a large wavenumber, too large to resolve in a computational code. An eddy viscosity is added to the model to make the dissipation occur at a much smaller wavenumber which can be resolved by the code.

[2] As opposed to the Favre averaged Navier-Stokes equations of the statistical methods.

The trick is to choose an eddy viscosity that does this “correctly.” Desired properties include:

- Smooth coupling of the resolved and unresolved scales
- Only need information from the resolved scales
- Eddy-viscosity is not overcome by numerical diffusive errors
- Correct behavior in “transition to turbulence” flow regimes
- No eddy-viscosity in laminar flow regimes
- A mechanism for the unresolved scales to affect the resolved scales

The original Smagorinsky eddy-viscosity model described below addressed all of these concerns but the last.

There are numerous general references for subgrid-scale modelling. The collection of articles edited by Dwoyer, Hussaini, and Voigts [29] include many. Specific reviews of LES are given by Herring [40], Rogallo and Moin [92], and most recently, Mason [73]. Much current work in LES is found in conference proceedings; see Galperin and Orszag [35], and Dannevik, Buckingham and Leith [25] for instance.

22.1 The Smagorinsky Eddy-Viscosity

The original use of the Smagorinsky eddy-viscosity was in a computation of atmospheric flow for a meteorological study [97]. The actual description of the model is a only a small part of the paper; quite a modest beginning for a method that is used so often. The original model added an eddy-viscosity to the incompressible Navier-Stokes equations:

$$\nu_T \sim \sqrt{\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)} \qquad \text{(EQ 58)}$$

with a turbulent thermal diffusivity proportional to this through a turbulent Prandtl number. The new (incompressible) stress tensor would be:

$$\tau_{ij} = -P\delta_{ij} + \rho(\nu + \nu_T)\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) \quad \text{(EQ 59)}$$

The development of Equation 58 merits some discussion. The slightly more general case of a compressible fluid, where the divergence of velocity is not equal to zero, will be presented. The basis for “deriving” Equation 58 is dimensional analysis of the possible contributing terms, and the assumption that there exists a balance between local turbulent energy production and dissipation.

The resolvable length scale is assumed to be λ; the resolvable wavenumber thus becomes $k \sim \frac{1}{\lambda}$. The velocity scale will come from the square root of turbulent kinetic energy:

$$K = \int_k^\infty E(k)\,dk \sim \varepsilon^{\frac{2}{3}}\lambda^{\frac{2}{3}} \quad \text{(EQ 60)}$$

This result can be derived from the well know Kolmogorov energy spectrum:

$$E(k) \sim \varepsilon^{\frac{2}{3}}k^{\frac{-5}{3}} \quad \text{(EQ 61)}$$

which in turn is derived from dimensional analysis. ε is the viscous dissipation rate, which is occurring at scales smaller than λ.

The dimensions of kinematic viscosity are that of a length times a velocity, so $\nu_T \sim \lambda\sqrt{K}$ is the correct dimensional choice. Equation 60 relates dissipation to turbulent kinetic energy,

$$\varepsilon \sim \frac{K^{\frac{3}{2}}}{\lambda} \quad \text{(EQ 62)}$$

so the set of relations is almost closed. All that remains is to find an expression for K in terms of resolved quantities.

The final component that Smagorinsky used is the assumption that there is a local balance between the production of turbulent kinetic energy, and its dissipation. The production is from the eddy shear stresses $\nu_T S_{ij}$ which has a corresponding work term $\nu_T S_{ij}\frac{\partial u_i}{\partial x_j}$. This work term is set equal to the turbulent kinetic energy K. The strain rate tensor for a compressible fluid is:

$$S_{ij} = \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3}\frac{\partial u_k}{\partial x_k}\delta_{ij} \tag{EQ 63}$$

The work term can be simplified by noting that

$$S_{ij}\frac{\partial u_i}{\partial x_j} = \frac{1}{2}S_{ij}S_{ij} = S^2 \tag{EQ 64}$$

Thus the production of turbulent kinetic energy is $K = \nu_T S^2$, which by the above assumption is set equal to the viscous dissipation $K = \nu_T S^2 = \varepsilon$. This final expression closes the set of equations, resulting in the equation for eddy viscosity

$$\nu_T = (C_S\lambda)^2 S \tag{EQ 65}$$

after the dimensionless constant C_S is included. This is identical to Equation 58, except that now the strain rate term includes the divergence of velocity, and the dimensionless constants are included. In practice, the length scale λ is typically set to the grid size.

The eddy-viscosity of Equation 58 has been used extensively. The previously mentioned reviews and conference proceedings give copious examples. This model has its deficiencies too (like all turbulence models!) which has elicited negative comments; see Kim, Moin and Moser [47] for one example. The model has been extended to remove some of these deficiencies, or to add additional capabilities. Germano, Piomelli, Moin and Cabot [37] extend it to handle boundary effects better. Leith [58][59] extends it to include compressibility, buoyancy, and stochastic backscatter. These latter are the extensions that will be used in this dissertation.

22.1.1 Eddy-viscosity implementation

The eddy-viscosity model used here is the standard Smagorinsky model, extended in a natural way for three-dimensional compressible flow. A stochastic backscatter term is added to the eddy-viscosity model, to account for the effect of subgrid-scale interactions upon the larger resolved scales. This backscatter term will be discussed in the next section.

The eddy-viscosity differs from Equation 58 in that terms are added to take into account buoyancy and compressive effects. The resultant eddy-viscosity is:

$$\nu_T = (C_S \Delta x)^2 \left[\sqrt{S^2 + B^2 + \left(C_D \frac{\partial u_k}{\partial x_k} \right)^2} - C_D \frac{\partial u_k}{\partial x_k} \right] \quad \text{(EQ 66)}$$

with a rate of strain term

$$S^2 = \frac{1}{2} \left[\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3} \frac{\partial u_k}{\partial x_k} \delta_{ij} \right]^2 \quad \text{(EQ 67)}$$

and a buoyancy term

$$B^2 = -\left(\frac{1}{Sc} \right) \left(\frac{\frac{\partial \rho}{\partial x_k}}{\rho} \right) \left(\frac{\frac{\partial p}{\partial x_k}}{\rho} \right) \quad \text{(EQ 68)}$$

which is set to zero if it is negative[3].

There are two additional terms in Equation 66 as compared to Equation 65. The first is the buoyancy term B^2. This term models the generation of turbulent kinetic energy at points where there are opposing density and pressure gradients. In a situation of hydrostatic balance, these gradients are of the same sign. In unstable regions where there are buoyancy forces, they will be of opposite sign.

The other addition term is the divergence term both inside and outside the square root. In the absence of the strain rate and buoyancy terms, this divergence term acts exactly the same as a Richtmyer - von Neumann artificial viscosity. In regions of compressive behav-

[3] There are two reasons for this: 1) To avoid a complex root of Equation 66; 2) Only opposing gradients of density and pressure should contribute to the generation of turbulent kinetic energy.

ior, the eddy viscosity is present. In regions of rarefaction, there is no additional eddy viscosity.

Equation 66 is used as an additional viscosity in the Navier-Stokes equations Equation 21 - Equation 23, by modifying the stress tensor (Equation 25) as follows:

$$\tau_{ij} = \left(-P - \frac{2}{3}(\mu + \mu_T)\frac{\partial u_k}{\partial x_k} \right)\delta_{ij} + (\mu + \mu_T)\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \quad \text{(EQ 69)}$$

The relationship between kinematic and dynamic viscosity is $\mu_T = \rho\nu_T$. The turbulent thermal diffusivity is then:

$$\kappa_T = \frac{\mu_T c_p}{Pr_T} \quad \text{(EQ 70)}$$

where the turbulent Prandtl number is taken to be $Pr_T = 0.7$ (equal to the Schmidt number Sc.) This gives a heat flux for the energy equation (Equation 23) of:

$$q_i = -\left(\frac{\mu\gamma}{Pr} + \frac{\mu_T\gamma}{Pr_T} \right)\frac{\partial}{\partial x_i}\hat{u} \quad \text{(EQ 71)}$$

In practice the molecular viscosity and thermal diffusivity are very small compared to their eddy counterparts; they are often set to zero.

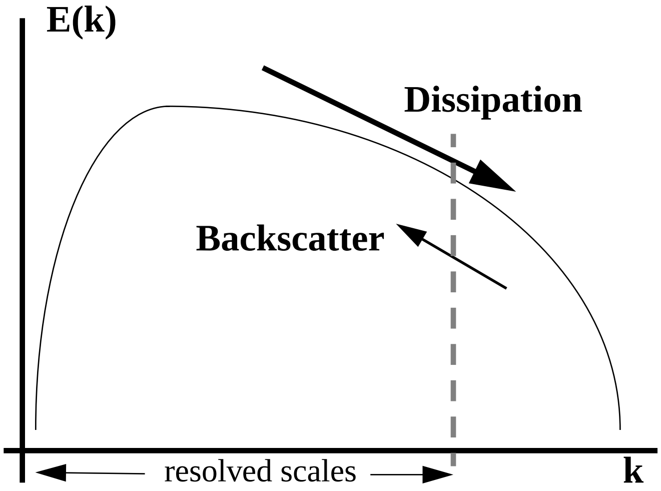
This model contains two adjustable coefficients, and the length scale Δx. This length scale is set to the computational grid size (assuming a uniform mesh.) The adjustable coefficients are the Smagorinsky coefficient C_S and the divergence coefficient C_D. The Smagorinsky coefficient is typically set to $C_S = 0.2$ This has been verified by DNS and predicted by renormalized group theory (RNG) [108] as being of the correct magnitude, if the cut-off length scale is set to the grid size. It is desired to have C_S as small a possible to avoid unnecessary diffusion, while yet large enough so that it still has the effect of smoothing out discontinuities and dissipating kinetic energy at resolved scales. Mason and Callen [70] and Mason and Thompson [71] give examples of large-eddy simulations done with various values of C_S. These computations give good comparisons of the effects of various magnitudes of C_S.

In the experience of both Leith [58][59] and this author, the numerical computation was very sensitive to stability problems when C_D took on significant (order of magnitude one or greater) values. It had insignificant effect below these levels. As it is a direct counterpart to the Richtmyer - von Neumann compressive artificial viscosity (Equation 84,) this should not be the case. Future work is needed to sort this difficulty out. In this dissertation, as in the work of Leith, C_D is set to zero, or restricted to values of less than or equal to one.

22.2 Stochastic Backscatter

The Smagorinsky model ignores one of the desirable properties listed above: that of the subgrid-scales being able to affect the resolved scales. It is now generally believed that this process can be important.

FIGURE 41. Dissipation and Backscatter



Eddy-viscosity models remove large-scale motion by turning it into internal energy through a viscosity-like term; this is the downward cascade of dissipated energy shown in Figure 41. While this allows the resolved scales to affect the subgrid scale turbulent energy level, there is no way for the small scales to affect the larger. In reality of course, these small scales do affect the larger through nonlinear interactions. The earliest subgrid-scale models ignored the backscatter except as it diminished the viscous damping.

However, there are cases where this small backscatter of energy from the smaller scales to the larger cannot be ignored. One example is in studies of transition to turbulence. Here the smallest motions by definition affect the unstable motion. A small perturbation that exists on a scale smaller than the resolved grid would start motion in a stationary unstable state. But since this motion is no longer resolved, the flow incorrectly remains stationary.

The need for backscatter terms in eddy-viscosity models has been known since their inception. Early papers that point this out include Fox and Lilly [31], and Herring [39]. Analytic studies of homogeneous turbulence have also shown the need for a backscattering term. Kraichnan [50], and Leslie and Quarini [63], are the basic references. Chasnov [21] showed in homogeneous turbulence computations, by comparisons with DNS, that LES performance was greatly improved by a stochastic backscatter term. Besides the aforementioned Leith papers [58][59] which deal with shear flows, there are other important works on non-homogeneous problems. Mason and Thompson [72] did a calculation of planetary boundary layers. Piomelli, Cabot, Moin and Lee [83] examine channel flow; they also do cross-comparisons with DNS.

22.2.1 Stochastic backscatter implementation

One way to implement backscatter is by adding a forcing term to the equations of motion. This forcing will model the effect of the small scales on the larger. Leith [58][59] suggests that this forcing be statistically time-white, spatially isentropic, and divergence free. It will only affect the scales just above the grid size. This simulates the assumed homogeneous, isotropic behavior of the turbulence that is occurring at the subgrid-scale. This random forcing (or acceleration) has zero mean and its variance comes from dimensional scaling arguments. All unspecified constants are lumped into one constant.

The stochastic backscatter model thus implemented is a random forcing vector potential

$$\phi_k = C_b (S\Delta t)^{\frac{3}{2}} \left(\frac{\Delta x}{\Delta t}\right)^2 g_k \qquad \text{(EQ 72)}$$

which will be added into the conservation of momentum equation (Equation 22) in conservative form:

$$\frac{\partial}{\partial t}\rho u_i + \frac{\partial}{\partial x_j}(\rho u_i u_j - \tau_{ij} + \varepsilon_{jik}\rho\phi_k) = \rho f_i \qquad \text{(EQ 73)}$$

There is one more adjustable coefficient in the stochastic backscatter model, $C_b = 0.2$. The Δx in the stochastic backscatter model is a length scale of the subgrid scale model. The stochastic backscatter model also uses a randomly generated three component unit gaussian field, g_k. It is the curl (via the tensor ε_{jik}) of the resultant vector field ϕ_k which generates the divergence free random forcing. Finally, S is the local scalar strain rate (Equation 67,) which is also used in the Smagorinsky eddy-viscosity.

Appendix C - Finite Difference Methods

23.0 Introduction

Finite difference methods [FDM] are discrete representations of continuous partial differential equations [PDE]; the intended use is the computational solution of the PDE. FDM are just one of the numerous methods for the numerical solutions of PDE. Other popular methods include:

- Finite Volume Methods [FVM]
- Finite Element Methods [FEM]
- Lattice Gas Methods
- Monte Carlo Simulations

In some cases it can be shown that these methods are equivalent. For instance, on uniform Cartesian grids the FDM and the FVM give rise to the same set of discrete equations to be solved. There are many good general references on the numerical solutions of PDE [33][42][64][99].References for their use on the equations of fluid dynamics are also plentiful [34][43][81].

This dissertation is concerned with a specific family of FDM, consisting of the so-called *Lax-Wendroff methods*. This refers to a second-order accurate, spatially centered, explicit FDM used to solve hyperbolic PDE. Second-order accurate means the discretization of the spatial derivatives are accurate to within terms of order Δx^2, while the temporal derivative are accurate to within Δt^2. In general the smaller the grid size (or time step) the more accurate the calculation.

Within the domain of finite difference methods, there are a number of ways to solve time dependent partial differential equations. The simplest are the explicit methods, which means that current and earlier time information is used to calculate the state at a later time. More specifically, to calculate the future value of a given spatial point, only current and past information of the point and its neighbors is required. This results in a linear system of algebraic equations to be solved that is purely diagonal in the future time value; thus only a trivial matrix inversion is required to solve for the next time step. These methods run quickly on a computer, and are usually easy to parallelize and to apply multigrid or adaptive gridding techniques. Their chief drawback is a result of causality; no time step can be larger than that such that the sound speed would allow information to travel across a cell in the grid. This is called the CFL condition, after the mathematicians Courant, Friedrichs, and Lewy [23] who first stated this constraint:

$$\Delta t \le \frac{\Delta x}{u_{max}} \qquad \text{(EQ 74)}$$

or

$$\varphi = \frac{u_{max}\Delta t}{\Delta x} \le 1 \qquad \text{(EQ 75)}$$

Δt is the size of the time step, Δx is the grid size, and u_{max} is a characteristic maximum velocity in the flow. The value φ is called the Courant number; the CFL condition is usually expressed as $\varphi \le 1$. The CFL condition states that the time step must be small enough that no signal can go across a cell in less than one time step. Notice that if a fine grid is required or if the characteristic velocity is very large then the time step must be very small (leading to long computation time.) If this limit is ignored (for instance by underestimating the maximum velocity) then the calculation becomes unstable, and a small scale fluctuation will “blow up” causing an overflow error on the computer. For this reason, the CFL condition is called a *stability condition.*

A simple explicit method uses spatially centered differences. It suffers from severe numerical dispersion and dissipation. Dispersive errors make waves travel at the wrong velocity; dissipative errors incorrectly damp waves. The upwind scheme (also called the donor cell method) is much better, but still has large numerical dissipation. Also, for problems more complicated than the linear one, it is sometimes difficult (or expensive) to determine which direction is upwind.

In an explicit algorithm, the presence of numerical dispersion is always bad. Dispersive errors cause the waves (or Fourier components of the initial conditions) to travel at an incorrect velocity. Numerical dispersion is especially serious for very large wave number components (i.e., sharp jumps, etc.) On the other hand, a little numerical dissipation can sometimes be a good thing. The damping can remove errors that may grow into instabilities. However, in and of themselves, the Euler equations have no diffusive or dissipative terms. So too much numerical dissipation will make the computation irrelevant and wrong, even if it is damping away instabilities.

As opposed to explicit methods, there are implicit methods. Implicit methods use both information from previous, current, and future times in the calculation. A centered algorithm uses equal weights on the current and future time step, while a fully implicit algorithm does all spatial differencing at the current time step. In either case, a linear system of simultaneous algebraic equations arise. This system is no longer has a diagonal matrix in the future time values as was the case in the explicit methods mentioned previously. Thus a non-trivial matrix inversion must be done. In a simple 1-D case, this may be as easy as solving a tri-diagonal linear system, for which reasonably simple, fast algorithms exists. But in multi-dimensional, multi-equation systems, the resultant matrix gives rise to is a complicated inversion problem which is computationally very costly.

A big advantage of the implicit methods is the ability to avoid the CFL condition. Effectively this means that a large time step can be made with out having to worry about stability problems. This can be a big help in many aerodynamic problems where a steady state solution is desired, and the evolution to that state is uninteresting. However, implicit methods suffer from defects other than the cost of matrix inversion. Specific algorithms can suffer from severe numerical dispersion and/or large numerical dissipation. Also, by taking large time steps, a subtle problem arises. The matrix inversion implies that all points on the grid know the state of all the other points instantaneously. This means that the numerical sound speed is very large; potentially greater than the speed of light! Non-physical results can thus be found, without ever seeing an instability problem.

Probably the best of the explicit algorithms are the Flux-Corrected algorithms, the TVD schemes, and the Lax-Wendroff methods. Flux-Corrected and TVD schemes are based on non-linear prescriptions that attempt to remove dispersive errors. However, they both require detailed knowledge of the eigenvalues of the system. While these are known for the Euler equations and are not to costly to compute, the situation for the Navier-Stokes equations is much more complicated. The Lax-Wendroff methods form a class of reasonably simple explicit algorithms; in a two-step form the need of knowing eigenvalues vanishes. From this point on, this paper will deal solely with the Lax-Wendroff methods.

The seminal text on the properties of FDM is the classic one by Richtmyer and Morton [91], while a more up to date source is Strikwerda [99]. Strikwerda also gives a very detailed explanation of dissipative and dispersive errors for scalar conservation equations. Three very good sources for comparisons of the various FDM on test problems (i.e., linear convection, convection-diffusion, Burger's equation, and the Riemann problem) are Hirsch [42][43], Fletcher [33][34], and Oran and Boris [78].

The method of analysis best suited for the non-linear conservation equations is differential approximation[1]. Klopfer and McRae carry this analysis out for the Euler equations with Lax-Wendroff type FDM in [48], which is based on the work of Warming and Hyett [105]. A more general analysis for numerous FDM for the Euler equations is carried out in the impressive and detailed Russian literature [109][110]. The general scheme of this analysis is put forth in Shokin's text [96]. The author of this dissertation has work in progress extending this analysis for the Lax-Wendroff methods to conservative systems with dissipation.

24.0 The Lax-Wendroff Method

The phase "Lax-Wendroff methods" has two separate meanings. The first meaning relates to a specific FDM for a scalar hyperbolic PDE. The second meaning is as a general term applied to all second order accurate (both spatial and temporal) central-differenced FDM

[1] So called in the Russian literature. The American literature calls this the "modified equation approach."

for a conservative system of hyperbolic PDE. These Law-Wendroff methods can be broken up into two distinct types: the single step methods; and the two-step methods. The single step methods advance one Δt in a single step. The two-step methods take two time advancements each of $\frac{1}{2}\Delta t$ to make a complete time cycle. Often these two-step methods are called predictor-corrector methods; the first half step generates a predictor value which is fed into the second corrector step. The single step algorithm was first introduced by Lax and Wendroff [55]; the two step algorithm of the type presented here was first introduced by Richtmyer and Morton [91] (it is sometimes known as the *Richtmyer algorithm*.) The complete development of the Lax-Wendroff method for the linear problem can be found in Strikwerda [99]. Equally good developments can be found in both Hirsch [42] and Oran and Boris [78].

24.1 The Richtmyer Scheme

The PDE to be solved is expressed in the conservative form with a source term:

$$\frac{\partial}{\partial t}\vec{u} + \frac{\partial}{\partial x}\vec{F}(\vec{u}) + \frac{\partial}{\partial y}\vec{G}(\vec{u}) + \frac{\partial}{\partial z}\vec{H}(\vec{u}) = \vec{K}(\vec{u}) \qquad \text{(EQ 76)}$$

Notice that the Navier-Stokes equations (Equation 21 - Equation 23) and the Euler equations (Equation 48 - Equation 50) have been put into this form.

The Richtmyer scheme [91] is a two step Lax-Wendroff method, set up for the conservative form of the equations of motion. The scheme presented here has been modified to include an external forcing term. The grid size is assumed a constant Δx, Δy and Δz; the time step is Δt.

The predictor step at time $t = (n+1)\Delta t$:

$$
\begin{aligned}
u^{n+1}{}_{ijk} = \frac{1}{6}(u^n_{i+1jk} + u^n_{i-1jk} + u^n{}_{ij+1k} + u^n{}_{ij-1k} + u^n{}_{ijk+1} + u^n{}_{ijk-1}) \\
-\frac{\Delta t}{2\Delta x}[F(u^n{}_{i+1jk}) - F(u^n{}_{i-1jk})] \\
-\frac{\Delta t}{2\Delta y}[G(u^n{}_{ij+1k}) - G(u^n{}_{ij-1k})] \\
-\frac{\Delta t}{2\Delta z}[H(u^n{}_{ijk+1}) - H(u^n{}_{ijk-1})] \\
+ \Delta t K(u^n{}_{ijk})
\end{aligned}
\tag{EQ 77}
$$

The corrector step at time $t = (n+2)\Delta t$:

$$
\begin{aligned}
u^{n+2}{}_{ijk} = u^n_{ijk} \\
-\frac{\Delta t}{\Delta x}[F(u^{n+1}{}_{i+1jk}) - F(u^{n+1}{}_{i-1jk})] \\
-\frac{\Delta t}{\Delta y}[G(u^{n+1}{}_{ij+1k}) - G(u^{n+1}{}_{ij-1k})] \\
-\frac{\Delta t}{\Delta z}[H(u^{n+1}{}_{ijk+1}) - H(u^{n+1}{}_{ijk-1})] \\
+ 2\Delta t K(u^{n+1}{}_{ijk})
\end{aligned}
\tag{EQ 78}
$$

Notice that the spatial grid is broken into a “checker-board” pattern. This means that the even cells have no influence on the odd cells. In pathological cases this can lead to a divergence of solutions on the even and odd grid points. Sometimes instead of using integral grid points, half grid points are used instead. In this case the time step for the predictor and corrector would be likewise reduced by a half. The choice is really a matter of style more than a mathematical difference.

24.2 The MacCormack Scheme

One way to get around the “checker-board” problem mentioned above is to slightly relax the necessity to always use central spatial differencing. MacCormack [69] developed a scheme much like the Richtmyer scheme that used forward differencing for either the predictor or corrector, and used backward differencing for the other. It is possible to show that for a scalar conservation law, the MacCormack, Richtmyer and Lax-Wendroff schemes are all equivalent and identical.

However, if say the forward spatial differences are always used for the predictor, and the backward differences are always used for the corrector then this will put a bias in the solution that is potentially as bad as the “checker-board” problem. Hence it is suggested that the direction of differencing of the spatial terms is alternated on the time steps. A very complete exposition on using the MacCormack scheme on both the Navier-Stokes equations and the Euler equations is given in Hirsch [43].

24.2.1 The Forward-Backward MacCormack Scheme

The predictor step uses forward differencing for spatial derivatives:

$$\begin{aligned}\bar{u}_{ijk} &= u^n_{ijk} \\ &-\frac{\Delta t}{\Delta x}\left[F\left({u^n}_{i+1jk}\right) - F\left({u^n}_{ijk}\right)\right] \\ &-\frac{\Delta t}{\Delta y}\left[G\left({u^n}_{ij+1k}\right) - G\left({u^n}_{ijk}\right)\right] \\ &-\frac{\Delta t}{\Delta z}\left[H\left({u^n}_{ijk+1}\right) - H\left({u^n}_{ijk}\right)\right] \\ &+ \Delta t K\left({u^n}_{ijk}\right)\end{aligned} \tag{EQ 79}$$

The corrector step uses backward differencing for spatial derivatives:

$$\begin{aligned}{u^{n+1}}_{ijk} &= \frac{1}{2}\left({u^n}_{ijk} + \bar{u}_{ijk}\right) \\ &-\frac{\Delta t}{2\Delta x}\left[F\left(\bar{u}_{ijk}\right) - F\left(\bar{u}_{i-1jk}\right)\right] \\ &-\frac{\Delta t}{2\Delta y}\left[G\left(\bar{u}_{ijk}\right) - G\left(\bar{u}_{ij-1k}\right)\right] \\ &-\frac{\Delta t}{2\Delta z}\left[H\left(\bar{u}_{ijk}\right) - H\left(\bar{u}_{ijk-1}\right)\right] \\ &+ \frac{\Delta t}{2} K\left(\bar{u}_{ijk}\right)\end{aligned} \tag{EQ 80}$$

24.2.2 The Backward-Forward MacCormack Scheme

The predictor step uses backward differencing for spatial derivatives:

$$\begin{aligned}\bar{u}_{ijk} &= u^n_{ijk} \\ &-\frac{\Delta t}{\Delta x}\left[F\left(u^n{}_{ijk}\right) - F\left(u^n{}_{i-1jk}\right)\right] \\ &-\frac{\Delta t}{\Delta y}\left[G\left(u^n{}_{ijk}\right) - G\left(u^n{}_{ij-1k}\right)\right] \\ &-\frac{\Delta t}{\Delta z}\left[H\left(u^n{}_{ijk}\right) - H\left(u^n{}_{ijk-1}\right)\right] \\ &+ \Delta t K\left(u^n{}_{ijk}\right)\end{aligned} \qquad \text{(EQ 81)}$$

The corrector step uses forward differencing for spatial derivatives:

$$\begin{aligned}u^{n+1}{}_{ijk} &= \frac{1}{2}\left(u^n{}_{ijk} + \bar{u}_{ijk}\right) \\ &-\frac{\Delta t}{2\Delta x}\left[F\left(\bar{u}_{i+1jk}\right) - F\left(\bar{u}_{ijk}\right)\right] \\ &-\frac{\Delta t}{2\Delta y}\left[G\left(\bar{u}_{ij+1k}\right) - G\left(\bar{u}_{ijk}\right)\right] \\ &-\frac{\Delta t}{2\Delta z}\left[H\left(\bar{u}_{ijk+1}\right) - H\left(\bar{u}_{ijk}\right)\right] \\ &+ \frac{\Delta t}{2} K\left(\bar{u}_{ijk}\right)\end{aligned} \qquad \text{(EQ 82)}$$

The code developed for this dissertation initially used the Richtmyer scheme. The final version incorporated the Forward/Backward + Backward/Forward MacCormack scheme.

24.3 The Weaknesses of the Lax-Wendroff Methods

First-order FDM suffer from excessive diffusive-type numerical truncation errors. A sharp interface is smeared out after a few time steps. This led to the development of the second-order methods such as the Lax-Wendroff methods. However, these methods suffer from dispersive errors. This means that if the initial condition was decomposed into its Fourier components, each of these components would travel at a slightly incorrect speed. This leads to a sharp interface developing an oscillatory behavior around the jump as the different Fourier components become spread out due to traveling at slightly different phase velocities.

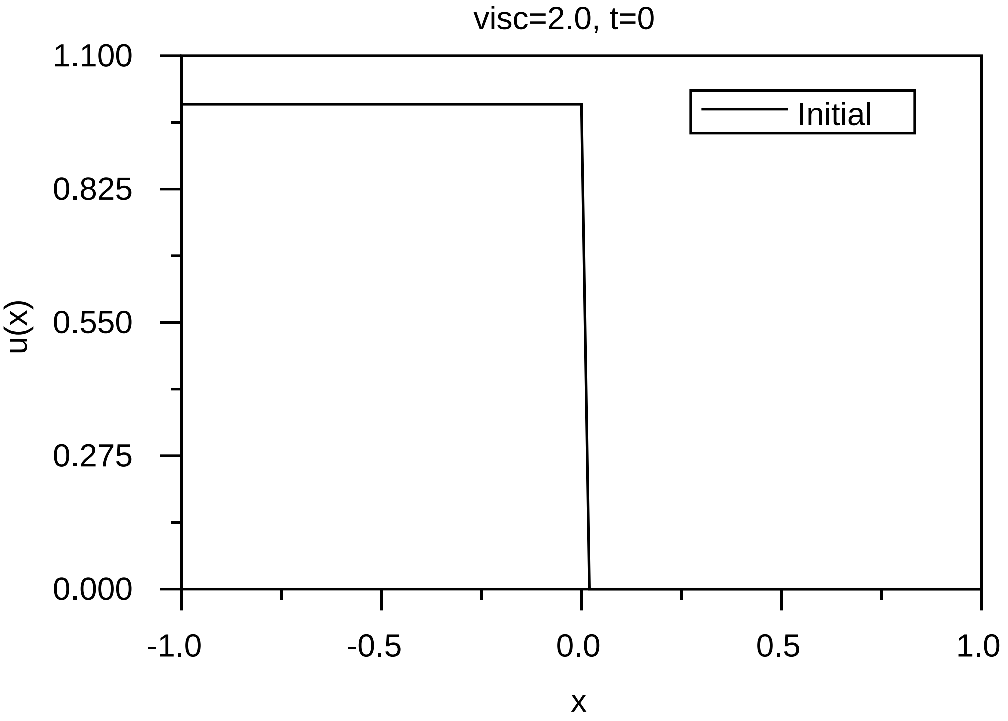
This observation is well documented in the literature. Hirsch [43] and Fletcher [34] for instance have detailed computed and analyzed examples. Strikwerda [99] is a good source of review for the group velocity concept developed by Trefethen [102]. Strikwerda also shows that in order to minimize these errors, the time step should be chosen as close to the CFL (or other stability limit) as is practical.

The following example will serve as an illustration. The inviscid 1-D Burgers' equation

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x}\left\{\frac{u^2}{2}\right\} = 0 \qquad \text{(EQ 83)}$$

is solved with the two-step MacCormack method. The initial condition (Figure 42) is the step function; there are 100 grid points. The analytic solution is compared to the MacCormack method (Figure 43) and the MacCormack method with Richtmyer-von Neumann artificial viscosity (Figure 44.)

FIGURE 42. Initial Condition, t=0



Burgers' equation is considered a reasonable model for the momentum equation of the one-dimensional, incompressible, inviscid Navier-Stokes equations. It captures the non-linear convective term. The MacCormack solution in Figure 43 shows a severe amount of oscillation on the high side of the discontinuity. This is the non-physical behavior that all Lax-Wendroff methods show at shocks and discontinuities. This is corrected to some extent by the addition of an Richtmyer-von Neumann artificial viscosity term, here with scaling constant $\alpha = 2.0$:

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x}\left\{\frac{u^2}{2}\right\} = \frac{\partial}{\partial x}\left\{\left[\alpha\Delta x\left|\frac{\partial u}{\partial x}\right|\right]\frac{\partial u}{\partial x}\right\} \quad \text{(EQ 84)}$$

The oscillations in Figure 44 are substantially less severe. However, the interface has lost its "sharpness," and is now more "smeared" out than the analytic solution. This artificial viscosity is discussed more fully in Richtmyer and von Neumann [77] and Richtmyer and Morton [91]. More complex eddy-viscosity models for the Navier-Stokes and Euler equations are discussed in "Appendix B - Turbulence Modelling" .

FIGURE 43. MacCormack Scheme at t=.36

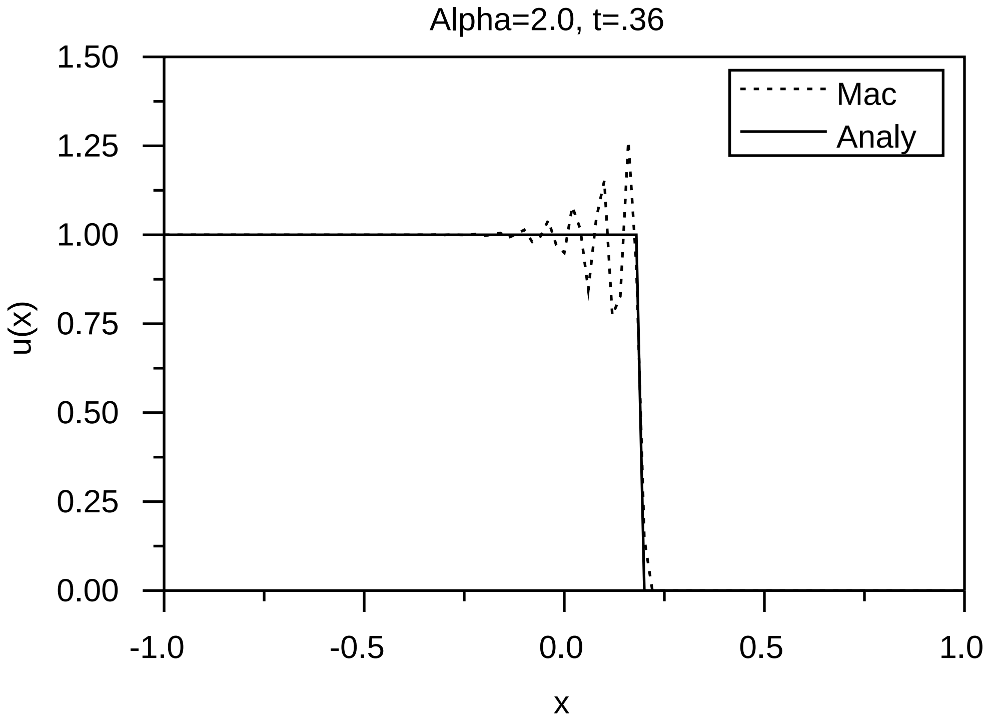
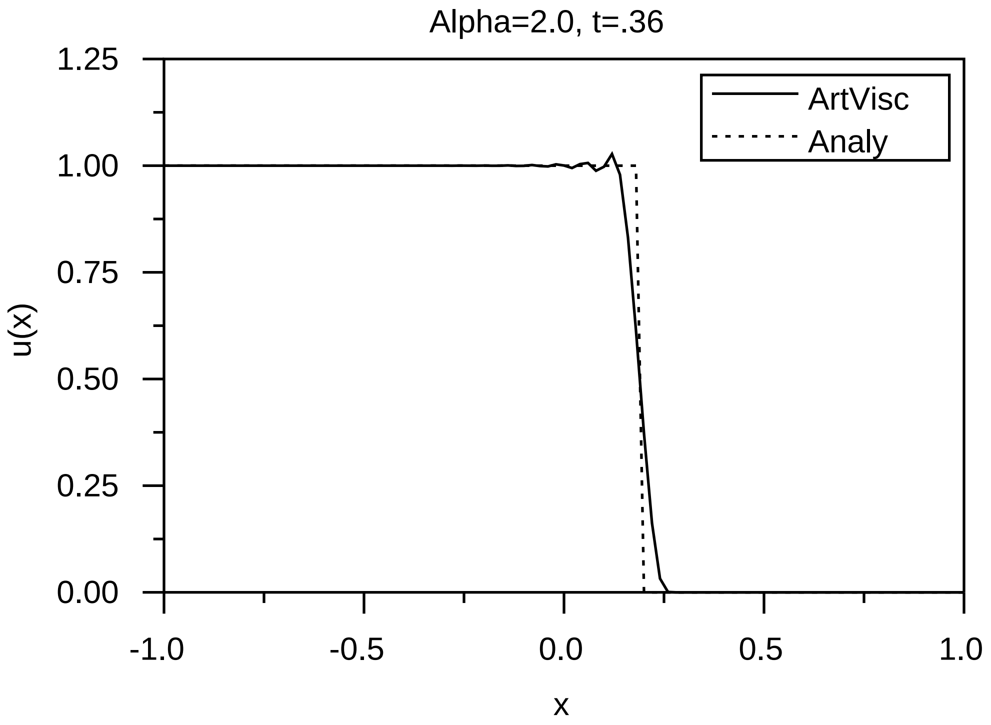


FIGURE 44. MacCormack Scheme with Artificial Viscosity at t=.36

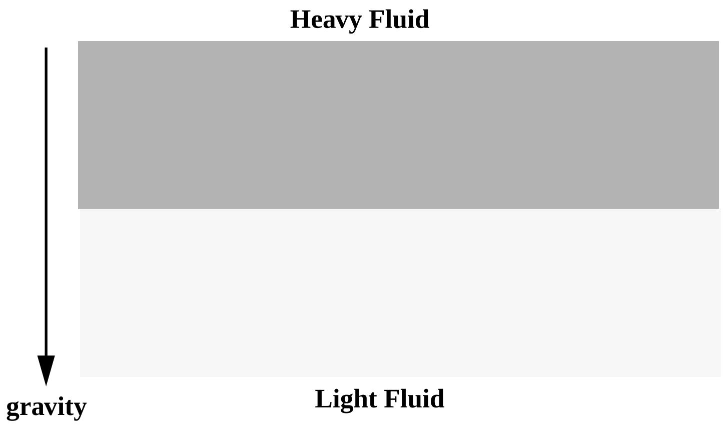


Appendix D - Linear Rayleigh-Taylor Theory

25.0 The Linear Theory

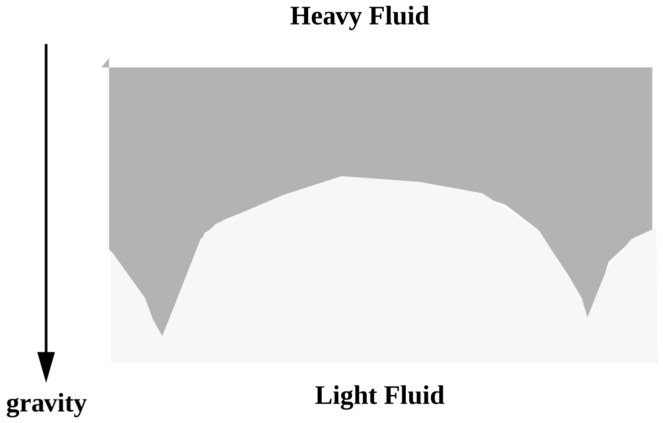
A fluid can be in one of three basic stability states. A *stable* fluid is one that will attempt to maintain the stable state. For instance, a light fluid on top of a heavy fluid is stable (think of the atmosphere on top of the ocean.) Even if this state is perturbed away from the stable state, the laws of physics attempt to bring it back to the original (or at least another stable) state. At the very least the disturbance will not grow; the disturbance will decay. A fluid can also be in an *unstable* state. In this case the fluid tends to move; it is attempting to reach some steady state or stable state. If the fluid is initially in a stationary state, the slightest perturbation will cause the fluid to start moving. This stationary initial condition of an unstable state is the base state that linear perturbation theory works upon. Finally, a fluid can be *neutrally* stable. A perturbation neither decays or grows. An example is a marble on a flat surface. The marble starts at rest; if it is touched it will roll. There is not a preferred position for the marble on the table, so the marble will roll forever in a straight line (assuming no friction, etc.)

FIGURE 45. The Rayleigh-Taylor Instability, initially



The situation in Figure 45 is unstable but stationary. If the interface between the two fluids is *perfectly* flat, then the heavy fluid will calmly sit on top of the light fluid forever. However, the smallest perturbation in the flat interface will disturb this balance. After a time the situation will be more like Figure 46. The fluids will continue to mix. Finally (assuming each fluid is finite in extent in the vertical direction) the light fluid will sit on top of the heavy fluid. In this state the two fluids would finally be stable.

FIGURE 46. The Rayleigh-Taylor Instability, later



Linear theory attempts to study the early phases of this process. First, a stationary base state is defined; this state trivially satisfies the equations of motion (for instance the Navier-Stokes or Euler equations.) Then this state has a perturbation added to represent an infinitesimal movement away from stationarity. The equations are resolved into unperturbed and perturbed components, but all terms which are of quadratic or higher order in the perturbations are discarded from the calculation. This leaves only the linear terms, which are then solved with the various techniques of partial differential equations. One typical method is to assume a sine expansion in some of the spatial dimensions, and perhaps exponential growth with time. This will leave a system of ordinary differential equations to be solved.

For a small enough amplitude of the perturbation, and for a small enough time the non-linear behavior of the original equations will be inconsequential, and thus the linear solution describes the initial behavior of the fluid.

The following derivation follows directly that of Gardner et. al [36], but with several typographical and other errors found there corrected. A solution is presented for the hydrostatic balance with a sharp density discontinuity at the interface. Then analytic results of the linear perturbation theory are presented, for both the 2-D and the 3-D cases.

25.1 The Hydrostatic Solution

The Euler equations are the starting point for the derivation of these solutions. Section 19.0 on page 92 details in full the Euler equations and required constitutive relations. The starting point here will be the vertical one-dimensional Euler equations:

$$\frac{\partial}{\partial t}\begin{bmatrix}\rho \\ \rho w \\ \rho e\end{bmatrix} + \frac{\partial}{\partial z}\begin{bmatrix}\rho w \\ \rho w^2 + P \\ (\rho e + P)\, w\end{bmatrix} = \begin{bmatrix}0 \\ \rho g \\ \rho w g\end{bmatrix} \quad \text{(EQ 85)}$$

Table 16, “Common Variables,” on page 84 gives a complete listing of the definitions of these variables. For the hydrostatic solution, it is assumed that the fluid is fully at rest, i.e. the vertical velocity component is $w = 0$. The z-axis is the vertical axis, and the constant gravity $\vec{g} = g$ points in the positive z-axis direction. While it is more common to think of the z-axis as pointing upwards, and gravity as pointing downwards, it is also possible to flip the z-axis to point downwards so that gravity and the z-axis can point in the same direction, while still keeping the familiar “gravity pointing to the ground.”[1] This is done to be consistent with the analysis of Read [87] and Youngs [112], who used a rocket shot downwards to get a strong “gravitational field.” The fluid is assumed to be isothermal with constant internal energy $\hat{u}_0$. This implies that the sound speed c^2 is constant. P_0 is the reference pressure at $z = 0$. With these assumptions, only the momentum equation remains which satisfies the hydrostatic balance condition:

$$\frac{\partial P}{\partial z} = \rho g \quad \text{(EQ 86)}$$

which after substituting in the equation of state (Equation 41)

$$dP = \rho g dz = \frac{Pg}{(\gamma - 1)\,\hat{u}} dz \quad \text{(EQ 87)}$$

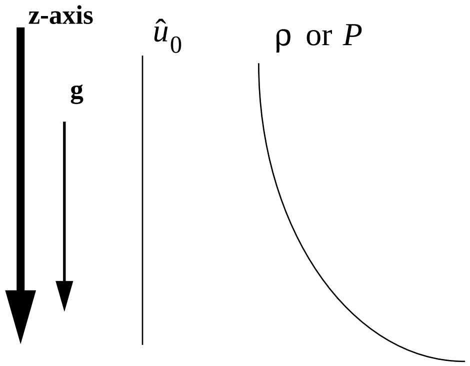
[1] This will be the approach used in this dissertation.

has the solution (by integration and specification of initial value) of:

$$P(z) = P_0 \exp\left(g \frac{z}{(\gamma - 1)\hat{u}_0}\right) = P_0 \exp\left(g\gamma \frac{z}{c^2}\right) \quad \text{(EQ 88)}$$

This is, of course, the hydrostatic solution for a compressible fluid with gravity. This solution is presented graphically in Figure 47.

FIGURE 47. Hydrostatic Fluid in a Gravity Field



It is possible to “stack” hydrostatic solutions upon one another. If a heavy fluid is placed under a lighter fluid this is of course a state of stability. The case of interest here is when the heavy fluid is place above the lighter. If the two fluids have the correct separate temperatures (i.e. internal energies) for the density jump at the interface, the fluids will be in a state of stationary instability. Let an “H” subscript represent the fluid state on the heavy side of the interface, and an “L” subscript for the light side. Assume that the two fluids interact at the interface without surface tension. Then, continuity of pressure at the interface (occurring at a value of $z = z_i$) implies that the pressure on either side of the interface must be equal:

$$P_H = (\gamma - 1)\rho_H \hat{u}_H = P_L = (\gamma - 1)\rho_L \hat{u}_L \quad \text{(EQ 89)}$$

If the density jump is by a factor $S \geq 1$ (where $S = \frac{\rho_H}{\rho_L}$,) then $\hat{u}_L = S\hat{u}_H$ must be true in order to ensure continuity of pressure[2]. This means that the lower lighter fluid must be hotter than the fluid above it for stationary interface to be ensured.

[2] The jump factor “S” used here should not be confused with the strain rate “S” of “Appendix B - Turbulence Modelling”

Thus the internal energy is a step function, with the jump occurring at the interface $z = z_i$:

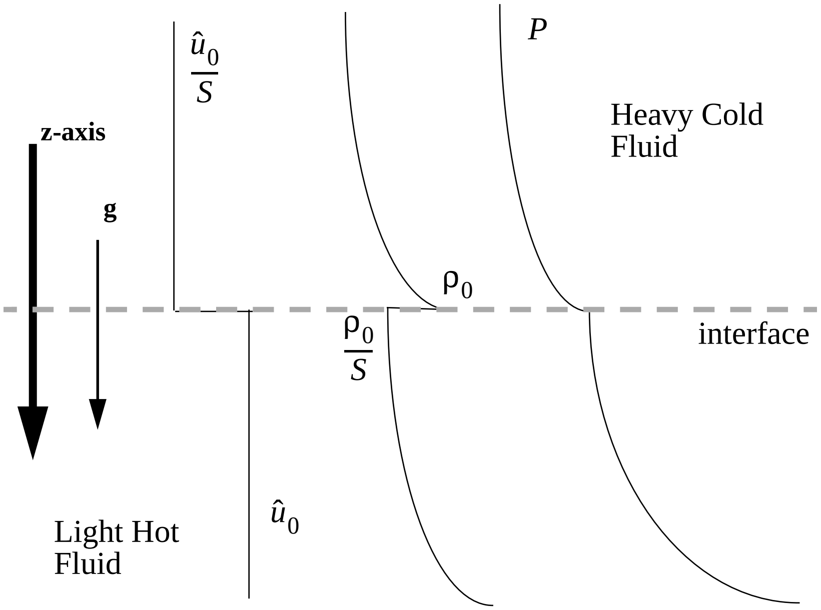
$$\hat{u}(z) = \begin{cases} \dfrac{\hat{u}_0}{S} \text{ if } z < z_i \\ \hat{u}_0 \text{ if } z > z_i \end{cases} \qquad \text{(EQ 90)}$$

where $\hat{u}_0$ is a reference energy level. The corresponding density is then

$$\rho(z) = \begin{cases} \rho_0 \exp\left(\dfrac{gS(z-z_i)}{(\gamma-1)\hat{u}_0}\right) \text{ if } z < z_i \\ \dfrac{\rho_0}{S} \exp\left(\dfrac{g(z-z_i)}{(\gamma-1)\hat{u}_0}\right) \text{ if } z > z_i \end{cases} \qquad \text{(EQ 91)}$$

where ρ_0 is the reference density at just above the jump. The pressure can be found by substituting Equation 90 and Equation 91 into the equation of state Equation 52. This solution is graphically presented in Figure 48.

FIGURE 48. Heavy Fluid Stably on Top of a Light Fluid



As long as the interface between the two fluids is perfectly flat, the two fluids will remain at rest and no motion will occur.

25.2 The Two-Dimensional Linear Solution

The next step is to add a perturbation to this flat interface in the horizontal direction. In order to do this the hydrostatic solution above is inserted into the two-dimensional Euler equations.

$$\frac{\partial}{\partial t}\begin{bmatrix}\rho \\ \rho u \\ \rho w \\ \rho e\end{bmatrix} + \frac{\partial}{\partial x}\begin{bmatrix}\rho u \\ \rho u^2 + P \\ \rho u w \\ (\rho e + P)\, u\end{bmatrix} + \frac{\partial}{\partial z}\begin{bmatrix}\rho w \\ \rho u w \\ \rho w^2 + P \\ (\rho e + P)\, w\end{bmatrix} = \begin{bmatrix}0 \\ 0 \\ \rho g \\ \rho w g\end{bmatrix} \quad \text{(EQ 92)}$$

For the hydrostatic case with no velocity in either the x- or z-directions, the solution is identical to that of Equation 88. Now consider the case where this stationary unstable state is perturbed by an infinitesimal amount. Then the solution would be

$$\begin{bmatrix}\rho \\ u \\ w \\ P\end{bmatrix} = \begin{bmatrix}\rho_0 \\ 0 \\ 0 \\ P_0\end{bmatrix} + \begin{bmatrix}\delta\rho \\ \delta u \\ \delta w \\ \delta P\end{bmatrix} \quad \text{(EQ 93)}$$

where the 'δf' represents the infinitesimal perturbation away from the stationary state of an arbitrary function 'f'. The subscript zero quantities 'f_0' represent the hydrostatic solution of Section 25.1 on page 122.

This solution is substituted back in to Equation 92 and any quantity of order δf^2 is set to zero[3]. This results in a set of first order coupled partial differential equations in the four perturbation quantities. The next step is to assume a separation of variables for these perturbation quantities. The time dependence is assumed to be exponential, and the horizontal spatial dependence is assumed to be sinusoidal. The vertical dependence is left indeterminate.

[3] Thus the name "linear theory," since only first order quantities remain.

$$\delta f\,(x, z, t) \;=\; \delta f\,(z)\,\exp\,(\sigma t)\,\exp\,(ikx) \qquad \text{(EQ 94)}$$

σ is the growth rate of the perturbation, and k is the wavenumber of the perturbation. The complex form of the spatial perturbation is used for the solution of the resultant ODE; only the real part of the solution need be retained. The thinking here is that the final solution for the perturbation will be proportional to $\varepsilon\cos\,(kx)$, where ε is the amplitude of the initial infinitesimal perturbation; the $\varepsilon i\sin\,(kx)$ term is dropped after the solution is completed

There is a slight abuse of notation here: the function δf is treated both as a function of x, z, and t, and also as a function of z alone. The use will be determined by context.

After the substitution of Equation 94 into the four PDE's of Equation 92 and the use of the linearized equation of state, there only remain 4 ODE's in the vertical variable z. By substitution these can be reduced to one second-order ODE, for instance in the pressure perturbation:

$$\frac{d^2}{dz^2}\delta P - \frac{\gamma g}{c^2}\frac{d}{dz}\delta P - \left(\frac{\sigma^2}{c^2} + k^2 + \frac{(\gamma - 1)\,g^2 k^2}{c^2 \sigma^2}\right)\delta P \;=\; 0 \qquad \text{(EQ 95)}$$

This is the classic Sturm-Liouville problem. For a specified wavenumber k, it is desired to find the eigenvalues σ that allow for a solution of the problem. All that needs to be specified are the boundary conditions.

It is assumed that the function is periodic in the horizontal x-direction. The interface is put in at $z = z_i$. A rigid, free slip, insulating lid is placed at the top and bottom vertical boundaries; $\left|z_i - z_b\right| \;=\; \delta z$ is the distance from the interface to either lid (i.e., for convenience, the interface is placed at the center of the vertical range.) Thus the vertical boundary conditions become:

$$\left.\frac{\partial}{\partial z}\hat{u}\right|_{z_i \pm \delta z} \;=\; 0 \qquad \text{(EQ 96)}$$

$$\left.\frac{\partial u}{\partial z}\right|_{z_i \pm \delta z} \;=\; 0 \qquad \text{(EQ 97)}$$

$$w|_{z_i \pm \delta z} = 0 \qquad \text{(EQ 98)}$$

Note that the sound speed is different above and below the interface. It is also possible for the specific heat ratio to differ above and below the boundaries, but here it is kept constant.

After much manipulation, the final solution for Equation 24 is

$$\begin{aligned}\delta P = -\left(\frac{(\gamma-1)g^2 + c^2\sigma^2}{\exp(\alpha_-(z_i - z_b)) - \exp(\alpha_+(z_i - z_b))}\rho_0\right)\varepsilon\exp(\sigma t)\exp(ikx) \\ \times\left(\frac{\exp(\alpha_-(z-z_b))}{(\gamma-1)g + c^2\alpha_-} - \frac{\exp(\alpha_+(z-z_b))}{(\gamma-1)g + c^2\alpha_+}\right)\exp\left(\frac{\gamma g}{c^2}(z-z_i)\right)\end{aligned} \qquad \text{(EQ 99)}$$

By substitution back into the other equations, we find the remaining perturbations to be:

$$\delta\rho = \frac{1}{(\gamma-1)g^2 + c^2\sigma^2}\left(\sigma^2\delta P + (\gamma-1)g\frac{d}{dz}\delta P\right) \qquad \text{(EQ 100)}$$

$$\delta w = \frac{1}{\sigma\rho_0}\left(g\delta\rho - \frac{d}{dz}\delta P\right) \qquad \text{(EQ 101)}$$

$$\delta u = \frac{-ik}{\sigma\rho_0}\delta P \qquad \text{(EQ 102)}$$

where

$$\alpha_\pm = -\frac{\gamma g}{2c^2} \pm \sqrt{\frac{\gamma^2 g^2}{4c^4} + \frac{\sigma^2}{c^2} + k^2 + \frac{(\gamma-1)g^2k^2}{c^2\sigma^2}} \qquad \text{(EQ 103)}$$

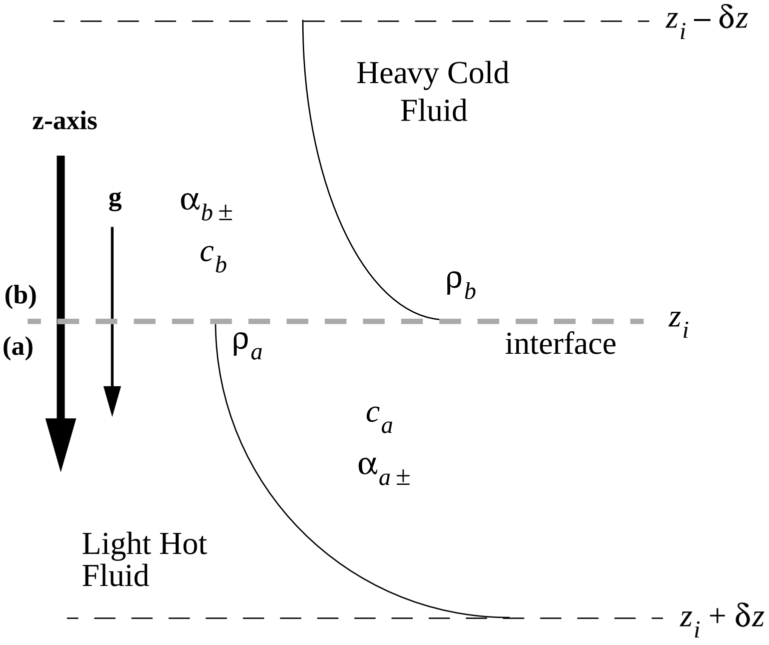
Note that the variable α takes on different values above and below the interface; the two different c^2 values of determine this. The “plus” and “minus” come from the two separate roots of the second order Equation 95. Thus there are actually four values of this constant α; in each half plane there are two values, one α_+ and the other α_-, each dependent on the thermodynamic values in said half plane. Also note that the above solution is complex. The real portion of each is taken as the final solution.

This gives the solution of the PDE, but the growth rate σ is yet to be determined. This can be done using the continuity of pressure at the interface. From this constraint it can be shown that

$$
\begin{aligned}
(\rho_b - \rho_a)\, g = & -\frac{(\gamma - 1)\, g^2 + c_a{}^2 \sigma^2}{\exp\,(-\alpha_{\text{a-}} |z_i - z_b|) - \exp\,(-\alpha_{\text{a+}} |z_i - z_b|)} \\
& \times \rho_a \left(\frac{\exp\,(-\alpha_{\text{a-}} |z_i - z_b|)}{(\gamma - 1)\, g + c_a{}^2 \alpha_{\text{a-}}} - \frac{\exp\,(-\alpha_{\text{a+}} |z_i - z_b|)}{(\gamma - 1)\, g + c_a{}^2 \alpha_{\text{a+}}} \right) \\
& + \frac{(\gamma - 1)\, g^2 + c_b{}^2 \sigma^2}{\exp\,(\alpha_{\text{b-}} |z_i - z_b|) - \exp\,(\alpha_{\text{b+}} |z_i - z_b|)} \\
& \times \rho_b \left(\frac{\exp\,(\alpha_{\text{b-}} |z_i - z_b|)}{(\gamma - 1)\, g + c_b{}^2 \alpha_{\text{b-}}} - \frac{\exp\,(\alpha_{\text{b+}} |z_i - z_b|)}{(\gamma - 1)\, g + c_b{}^2 \alpha_{\text{b+}}} \right)
\end{aligned}
\qquad \text{(EQ 104)}
$$

The subscripts represent the values of these variables just above (larger value of z) (“a”) or below (smaller value of z) (“b”) the interface. The following picture should sort out the location of values.

FIGURE 49. Location of Values for Linear Perturbation Theory Functions

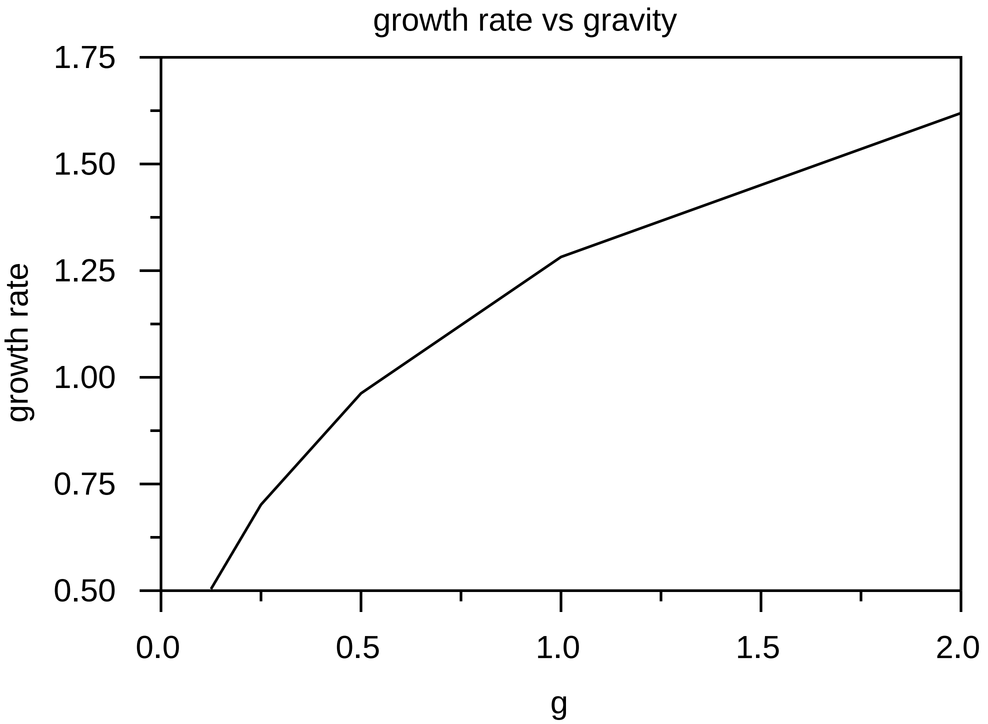


By solving the non-linear Equation 104 with known values for every other term, the growth rate can be recovered. Equation 104 an extremely non-linear equation, it must be solved by numerical techniques. Simplifications and attempts at non-dimensionalizing this equation are only partially successful; this author found it easier to solve this equation with some mathematical package with both numerical and symbolic manipulation[4]. The case of infinite boundaries does help the situation slightly; for instance see Bernstein and Book [16] for particulars.

TABLE 18. Growth Rate as a Function of Gravity

g	σ
0.125	0.5038227000
0.25	0.7016423213
0.5	0.9624588156
1.0	1.282123386
2.0	1.619374429

FIGURE 50. Growth Rate as a Function of Gravity



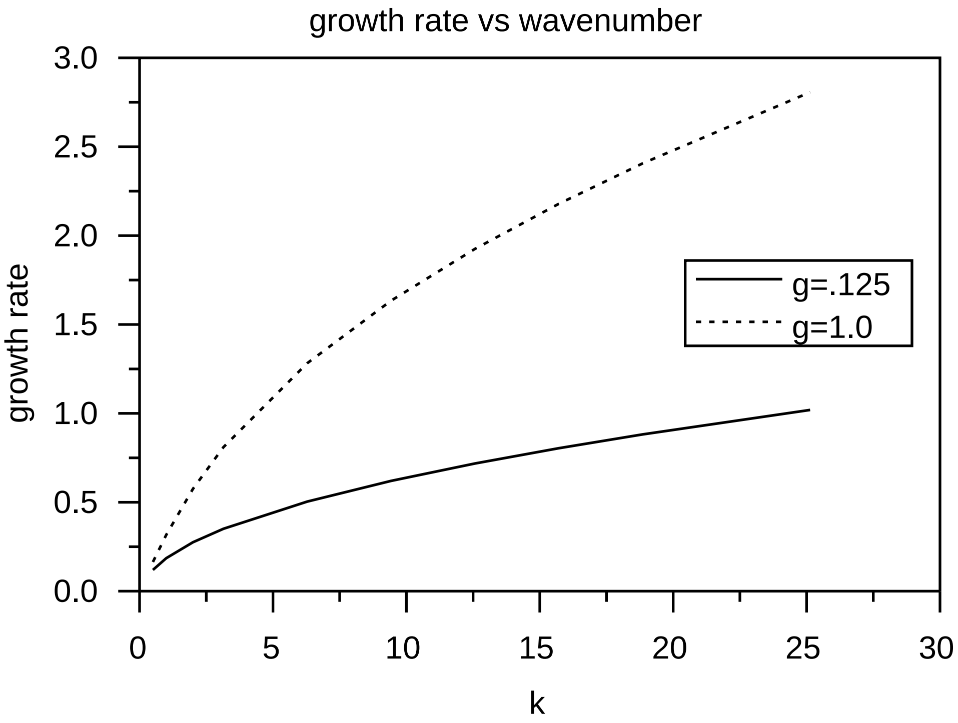
[4] This author uses Maple V R2 on a Macintosh.

Table 18 (and the corresponding Figure 50) is based on Equation 104. The values are calculated for $k = 2\pi$, $\gamma = \frac{5}{3}$, with an Atwood number[5] of $A = \frac{1}{3}$ (this corresponds to a density jump of 2.) The growth rate was only weakly dependent on the distance to the boundary. Table 18 holds roughly in the range $2 \le |z_i - z_b| \le 8$. A third-order polynomial fit to this data is:

$$\sigma = 0.265038g^3 - 1.11751g^2 + 1.83599g + 0.297208 \qquad \text{(EQ 105)}$$

Table 19 (and the corresponding Figure 51) is also based on Equation 104. The values are calculated for $g = 0.125$ and $g = 1.0$, $\gamma = \frac{5}{3}$, with an Atwood number of $A = \frac{1}{3}$. Table 19 also holds in the range $2 \le |z_i - z_b| \le 8$

FIGURE 51. Growth Rate as a Function of Wavenumber



For $g = 0.125$ the third-order polynomial curve fit to the data of Table 19 is:

$$\sigma = 6.501\times10^{-5}k^3 - 0.00034926k^2 + 0.0833159k + 0.103538 \qquad \text{(EQ 106)}$$

[5] $A = \frac{\rho_H - \rho_L}{\rho_H + \rho_L}$ is the Atwood number; it is a common measure of the density jump.

while for $g = 1.0$ it is:

$$\sigma = 0.00018165k^3 - 0.0100055k^2 + 0.245266k + 0.0883437 \qquad \text{(EQ 107)}$$

TABLE 19. Growth Rate as a Function of Wavenumber

k	σ (g = .125)	σ (g = 1.0)
0.5	0.1193565846	0.1640998325
1.0	0.1854888438	0.3167870972
2.0	0.2750710993	0.5752478157
π	0.3508211607	0.8096872144
2 π	0.5038227000	1.282123386
3 π	0.6202343300	1.633808378
4 π	0.7180299770	1.924917631
5 π	0.8040228568	2.178453725
6 π	0.8816706371	2.405877973
8 π	1.019377663	2.806569285

One of the easier simplifications of Equation 104 relies on the properties of the $\alpha_{\pm}$ term. This term will be either positive or negative, and thus one of the exponential terms in Equation 104 will be very large compared to the other. This also holds in the limit of $|z_i - z_b|$ becoming large. This allows a simplification of Equation 104 to

$$(\rho_b - \rho_a)\, g = -\frac{(\gamma - 1)\, g^2 + c_a^{\ 2}\sigma^2}{(\gamma - 1)\, g + c_a^{\ 2}\alpha_{\mathrm{a-}}}\rho_a - \frac{(\gamma - 1)\, g^2 + c_b^{\ 2}\sigma^2}{(\gamma - 1)\, g + c_b^{\ 2}\alpha_{\mathrm{b+}}}\rho_b \qquad \text{(EQ 108)}$$

This is still a non-linear equation, since the $\alpha_{\pm}$ depend on a square root of σ. A closed formed explicit solution for σ as a function of the other variables is the real positive root of a polynomial of degree four. This author was not able to work this root out analytically (with the assistance of Maple V,) thus once again a numerical solution must be used. Since this is the case, in this dissertation the more general Equation 104 will be used to determine the growth rates. However, for “on the fly” calculations within a computer code, the simplified form of Equation 108 may be a better choice.

25.3 The Three-Dimensional Linear Solution

The three-dimensional linear solution can be obtained from the two-dimensional solution easily through the “superposition of solutions” principle. The procedure is outlined below.

For a given wave-number k_x and amplitude ε_x in the x-direction, the corresponding growth rate σ_x is determined. The perturbations in the x-direction are then calculated using Equation 99 through Equation 102. The exact same procedure is then again carried out for the y-direction, this time using wave-number k_y and amplitude ε_y . Finally the two separate perturbations are added together to form the final state (typically used as an initial condition for a computation:)

$$\begin{bmatrix} \rho \\ u \\ v \\ w \\ P \end{bmatrix} = \begin{bmatrix} \rho_0 \\ 0 \\ 0 \\ 0 \\ P_0 \end{bmatrix} + \begin{bmatrix} \delta\rho_x \\ \delta u_x \\ 0 \\ \delta w_x \\ \delta P_x \end{bmatrix} + \begin{bmatrix} \delta\rho_y \\ 0 \\ \delta v_y \\ \delta w_y \\ \delta P_y \end{bmatrix} \qquad \text{(EQ 109)}$$

From these five equations, the variables used in the computation can then be recovered (i.e. momentum, total energy, etc.)

Appendix E - Bibliography

[1] **CM-5 Technical Summery**, Thinking Machines Corp., Nov. 1992

[2] **CM-5 User's Guide V7.2**, Thinking Machines Corp., Aug. 1992

[3] **Getting Started in CM Fortran**, Thinking Machines Corp., Jan. 1993

[4] **CM Fortran User's Guide V2.0 Beta**, Thinking Machines Corp., Jan. 1993

[5] **CM Fortran Reference Manual V2.0 Beta**, Thinking Machines Corp., Dec.1992

[6] **CM Fortran Utility Library Reference Manual V2.0 Beta**, Thinking Machines Corp., Jan. 1993

[7] **CM-5 I/O System Programming Guide CMOST V7.2**, Thinking Machines Corp., Sept. 1993

[8] **CMSSL for CM Fortran, CM-5 Edition V3.1**, Thinking Machines Corp., June 1993

[9] **CMMD User's Guide V3.0**, Thinking Machines Corp., May 1993

[10] **CMMD Reference Manual V3.0**, Thinking Machines Corp., May 1993

[11] **DPEAC Reference Manual CMOST V7.1**, Thinking Machines Corp., March 1993

[12] **VU Programmer's Handbook CMOST V7.1**, Thinking Machines Corp., Aug. 1993

[13] Paul A.K. Amala and Garry H. Rodrigue, *Programming Models for Three-Dimensional Hydrodynamics on the CM-5, Part I*, Computers in Physics, 8 (2), 1994

[14] Paul A.K. Amala and Garry H. Rodrigue, *Programming Models for Three-Dimensional Hydrodynamics on the CM-5, Part II*, Computers in Physics, 8 (3) , 1994

[15] Ian G. Angus, Geoffrey C. Fox, Jai Sam Kim, and David W. Walker, **Solving Problems on Concurrent Processors Vol. 2**, Prentice Hall, Englewood Cliffs, NJ, 1990

[16] I. B. Bernstein and D. L. Book, *Effect of compressibility on Rayleigh-Taylor instability*, Physics of Fluids, 26 (2), 1983

[17] D. L. Book and I. B. Bernstein, *Linear Theory of the Rayleigh-Taylor Instability at the Interface Between Two Compressible Media*, Naval Research Laboratory Memorandum Report 4880, Sept. 1982

[18] D. L. Book, *Rayleigh-Taylor Instability in Compressible Media*, Naval Research Laboratory Memorandum Report 5373, Aug. 1984

[19] Peter Bradshaw, Tuncer Cebeci and James H. Whitelaw, **Engineering Calculation Methods for Turbulent Flow**, Academic Press, London, 1981

[20] S. Chandrasekhar, **Hydrodynamic and Hydromagnetic Stability**, Dover, New York, 1981

[21] Jeffrey R. Chasnov, *Simulation of the Kolmogorov inertial subrange using an improved subgrid model*, Physics of Fluids (A), 3 (1), 1991

[22] A. J. Chorin and J. E. Marsden, **A Mathematical Introduction to Fluid Mechanics**, Springer-Verlag, New York, 1979

[23] R. Courant, K. O. Friedrichs, and H. Lewy, *Uber die partiellen Differenzengleichungen der mathematischen Physik*, Mathematics Annual, 100, 1928

[24] William P. Dannevik, Victor Yakhot, and Steven A. Orszag, *Analytic theories of turbulence and the* ε *expansion*, Physics of Fluids, 30 (7), 1987

[25] W. P. Dannevik, A. C. Buckingham and C. E. Leith (editors), **Advances in Compressible Turbulent Mixing**, DOE Conference Proceedings Conf-8810234, 1992, (from the International Workshop on the Physics of Compressible Turbulent Mixing, Princeton University, 1988)

[26] Andrzej J. Domaradzki and Robert S. Rogallo, *Local energy transfer and nonlocal interactions in homogeneous, isotropic turbulence*, Physics of Fluids (A), 2 (3), 1990

[27] P. G. Drazin and W. H. Reid, **Hydrodynamic Stability**, Cambridge Univ. Press, Cambridge, 1981

[28] F. Durst, B. E. Launder, F. W. Schmidt and J. H. Whitelaw (editors), **Turbulent Shear Flows I, Selected Papers from the First International Symposium on Turbulent Shear Flows** (Apr. 1977), Springer-Verlag, Berlin, 1979

[29] D. L. Dwoyer, M. Y. Hussaini and R. G. Voigt (editors), **Theoretical Approaches to Turbulence**, Springer-Verlag, New York, 1985

[30] Harley Flanders, **Differential Forms with Applications to the Physical Sciences**, Dover, New York, 1989

[31] D. G. Fox and D. K. Lilly, *Numerical simulation of turbulent flows*, Reviews of Geophysics and Space Physics, 10 (1), 1972

[32] Geoffrey C. Fox, Mark A. Johnson, Gregory A. Lyzenga, Steve W. Otto, John K. Salmon, and David W. Walker, **Solving Problems on Concurrent Processors Vol. 1**, Prentice Hall, Englewood Cliffs, NJ, 1988

[33] C. A. J. Fletcher, **Computational Techniques for Fluid Dynamics Vol. 1**, Springer-Verlag, Berlin, 1988

[34] C. A. J. Fletcher, **Computational Techniques for Fluid Dynamics Vol. 2**, Springer-Verlag, Berlin, 1988

[35] Boris Galperin and Steven A. Orszag (editors), in **Large Eddy Simulations Complex Engineering and Geophysical Flows**, Cambridge Univ. Press, Cambridge, 1993

[36] C. L. Gardner, J. Glimm, O. McBryan, R. Menikoff, D. H. Sharp, and Q. Zhang, *The dynamics of bubble growth for Rayleigh-Taylor unstable interfaces*, Physics of Fluids, 31

(3), 1988

[37] Massimo Germano, Ugo Piomelli, Parviz Moin and William H. Cabot, *A dynamic subgrid-scale eddy viscosity model*, Physics of Fluids (A), 3 (7), 1991

[38] S. G. Glendinning, S. V. Weber, P. Bell, L. B. DaSilva, S. N. Dixit, M. A. Henesian, D. R. Kania, J. D. Kilkenny, H. T. Powell, R. J. Wallace, P. J. Wegner, J. P. Knauer and C. P. Verdon, *Laser-driven planar Rayleigh-Taylor instability experiments*, Physical Review Letters, 69 (8), 1992

[39] J. R. Herring, *Statistical turbulence theory and turbulence phenomenology*, in **Proceedings of the Langley Working Conference on Free Turbulent Shear Flows**, NASA SP 321, Langley Research Center, VA, 1973

[40] J. R. Herring, *Subgrid scale modeling - An introduction and overview*, in F. Durst, B. E. Launder, F. W. Schmit J. H. Whitelaw (editors), **Turbulent Shear Flows I**, Springer-Verlag, New York, 1979

[41] W. Daniel Hillis and Bruce M. Boghosian, *Parallel Scientific Computation*, Science, 261, 13 Aug. 1993

[42] C. Hirsch, **Numerical Computation of Internal and External Flows Vol. 1**, John Wiley & Sons, Chichest, 1988

[43] C. Hirsch, **Numerical Computation of Internal and External Flows Vol. 2**, John Wiley & Sons, Chichest, 1990

[44] R. W. Hockney and C. R. Jesshope, **Parallel Computers 2**, Adam Hilger, Bristol, 1988

[45] Kiyosi Horiuti, *The role of the Bardina model in large eddy simulation of turbulent channel flow*, Physics of Fluids (A), 1 (2), 1989

[46] Kai Hwang and Faye A. Briggs, **Computer Architecture and Parallel Processing**, McGraw-Hill Book Comp., New York, 1984

[47] J. Kim, P. Moin and R. Moser, *Turbulence statistics in fully developed channel flow at low Reynolds number*, Journal of Fluid Mechanics, 177, 1987

[48] Goetz H. Klopfer and Davis S. McRae, *Nonlinear truncation error analysis of finite difference schemes for the Euler equations*, The American Institute of Aeronautics and Astronautics Journal, 21 (4), 1983

[49] Robert H. Kraichnan, *Inertial-range transfer in two- and three-dimensional turbulence*, Journal of Fluid Mechanics, 47 (3), 1971

[50] Robert H. Kraichnan, *Eddy viscosity in two and three dimensions*, Journal of the Atmospheric Sciences, 33, 1976

[51] Kenneth K. Kuo, **Principles of Combustion**, Wiley Interscience, New York, 1986

[52] Sir Horace Lamb, **Hydrodynamics** (sixth edition), Dover, New York, 1945

[53] L. D. Landau and E. M. Lifshitz, **Fluid Mechanics**, Pergamon Press, Oxford, 1975

[54] P. D. Lax and R. D. Richtmyer, *Survey of the stability of linear finite difference equations*, Communications on Pure and Applied Mathematics, 9, 1956

[55] P. D. Lax and B. Wendroff, *Systems of conservation laws*, Communications on Pure and Applied Mathematics, 13, 1960

[56] C. E. Leith and R. H. Kraichnan, *Predictability of turbulent flows*, Journal of the Atmospheric Sciences, 29 (6), 1972

[57] C. E. Leith, **Statistical Hydrodynamics of the Atmosphere**, Class notes for Courant Institute of Mathematical Sciences, NYU, Fall 1981

[58] C. E. Leith, *Stochastic backscatter in a subgrid-scale model: Plane shear mixing layer*, Physics of Fluids (A), 2 (3), 1990

[59] C. E. Leith, *Stochastic Backscatter Formulation for Three-Dimensional Compressible Flows*, Boris Galerpin and Steven A. Orszag (editors) in **Large Eddy Simulations Complex Engineering and Geophysical Flows**, Cambridge Univ. Press, Cambridge, 1993

[60] A.Lerat and R. Peyret, *The problem of spurious oscillations in the numerical solution of the equations of gas dynamics*, R. D. Richtmyer (editor) in **Fourth International Conference on Numerical Methods in Fluid Dynamics**, 1975

[61] M. Lesieur, **Turbulence in Fluids** (second revised edition), Kluwer Academic Publishers, Dordrecht, 1990

[62] D. C. Leslie, **Developments in the Theory of Turbulence**, Clarendon Press, Oxford, 1973

[63] D. C. Leslie and G. L. Quarini, *The application of turbulence theory to the formulation of subgrid modelling procedures*, Journal of Fluid Mechanics, 91, 1979

[64] Randall J. LeVeque, **Numerical Methods for Conservation Laws**, Birkhauser Verlag, Basel, 1992

[65] X. L. Li, *Study of three-dimensional Rayleigh-Taylor instability in compressible fluids through level set method and parallel computation*, Physics of Fluids (A), 5 (8), 1993

[66] D. K. Lilly, *A proposed modification of the Germano subgrid-scale closure method*, Physics of Fluids (A), 5 (5), 1992

[67] Lyle N. Long, *A three-dimensional Navier-Stokes method for the Connection Machine*, Horst D. Simon (editor) in **Proceedings of the Conference on Scientific Applications of the Connection Machine** (second edition), World Scientific, Singapore, 1992

[68] L. N. Long, M. Kamon, T. S. Chyczewski and J. Myczkowski, *A deterministic parallel algorithm to solve a model Boltzmann equation (BGK)*, Computing Systems in

Engineering, 3 (1-4), 1992

[69] R. W. MacCormack, *The effect of viscosity in hypervelocity impact cratering*, Technical Report 69-354, The American Institute of Aeronautics and Astronautics, 1969

[70] P. J. Mason and N. S. Callen, *On the magnitude of the subgrid-scale eddy coefficient in large-eddy simulations of turbulent channel flow*, Journal of Fluid Mechanics, 162, 1986

[71] P. J. Mason and D. J. Thomson, *Large-eddy simulation of the neutral-static-stability planetary boundary layer*, Quarterly Journal of the Royal Meteorological Society, 113, 1987

[72] P. J. Mason and D. J. Thomson, *Stochastic backscatter in large-eddy simulations of boundary layers*, Journal of Fluid Mechanics, 242, 1992

[73] P. J. Mason, *Large-eddy simulation: A critical review of the technique*, Quarterly Journal of the Royal Meteorological Society, 120, 1994

[74] Richard E. Meyer, **Introduction to Mathematical Fluid Dynamics**, Dover, New York, 1982

[75] A. S. Monin and A. M. Yoglom, **Statistical Fluid Mechanics Vol. 1**, MIT Press, Cambridge MA, 1971

[76] A. S. Monin and A. M. Yoglom, **Statistical Fluid Mechanics Vol. 2**, MIT Press, Cambridge MA, 1975

[77] J. Von Neumann and R. D. Richtmyer, *A method for the numerical calculation of hydrodynamic shocks*, Journal of Applied Physics, 21, 1950

[78] Elaine S. Oran and Jay P. Boris, **Numerical Simulation of Reactive Flow**, Elsevier, New York, 1987

[79] Steven A. Orszag, *Lectures on the statistical theory of turbulence*, in R. Balian and J. L. Puebe (editors), **Fluid Dynamics 1973: Les Houches Summer School of Theoretical Physic**s, Gordon and Breach, New York, 1977

[80] R. L. Panton, **Incompressible Flow**, Wiley Interscience, New York, 1984

[81] Roger Peyret and Thomas D. Taylor, **Computational Methods for Fluid Flow**, Springer-Verlag, Berlin, 1983

[82] Ugo Piomelli, Parviz Moin, and Joel H. Ferziger, *Model consistency in large eddy simulation of turbulent channel flows*, Physics of Fluids, 31 (7), 1988

[83] Ugo Piomelli, William H. Cabot, Parviz Moin, and Sangsan Lee, *Subgrid-scale backscatter in turbulent and transitional flows*, Physics of Fluids (A), 3 (7), 1991

[84] Ravi Ponnusamy, Rajeev Thakur, Alok Choudhary, Kishore Velamakanni, Zeki Bozkus, and Geoffrey Fox, *Experimental performance evaluation of the CM-5*, Journal of Parallel and Distributed Computing, 19, 1993

[85] Lord Rayleigh, *Investigation of the character of the equilibrium of an incompressible heavy fluid of variable density*, Proceedings of the London Mathematical Society, 14, 1883

[86] Lord Rayleigh, **Scientific Papers** Vol. II, Cambridge, 1900

[87] K. I. Read, E*xperimental investigation of turbulent mixing by Rayleigh-Taylor instability*, Physica D, 12, 1984

[88] B. A. Remington, S. W. Haan, S. G. Glendinning, J. D. Kilkenny, D. H. Munro and R. J. Wallace, *Large growth Rayleigh-Taylor experiments using shaped laser pulses*, Physical Review Letters, 67 (23), 1991

[89] B. A. Remington, S. W. Haan, S. G. Glendinning, J. D. Kilkenny, D. H. Munro and R. J. Wallace, *Large growth, planar Rayleigh-Taylor experiments on Nova*, Physics of Fluids (B), 4 (4), 1992

[90] B. A. Remington, S. V. Weber, S. W. Haan, J. D. Kilkenny, S. G. Glendinning, R. J. Wallace, W. H. Goldstein, B. G. Wilson and J. K. Nash, *Laser-driven hydrodynamic instability experiments*, Physics of Fluids (B), 4 (4), 1992

[91] Robert D. Richtmyer and K. W. Morton, **Difference Methods for Initial-value Problems**, (second edition), Interscience Publishers, New York, 1967

[92] R. S. Rogallo and P. Moin, *Numerical simulation of turbulent flows*, Annual Review of Fluid Mechanics, 16, 1984

[93] Hermann Schlichting, **Boundary-Layer Theory** (seventh edition), McGraw-Hill, New York, 1987

[94] Lee A. Segel, **Mathematics Applied to Continuum Mechanics**, Dover, New York, 1987

[95] D. H. Sharp, *An overview of the Rayleigh-Taylor instability*, Physica D, 12, 1984

[96] Yu. I. Shokin, **The Method of Differential Approximation**, Springer-Verlag, Berlin, 1983

[97] J. Smagorinsky, *General circulation experiments with the primitive equations*, Monthly Weather Review, 91 (3), 1963

[98] M. M. Stanisic, **The Mathematical Theory of Turbulence**, Springer-Verlag, New York, 1985

[99] John C. Strikwerda, **Finite Difference Schemes and Partial Differential Equations**, Wadsworth & Brooks/Cole, Pacific Grove CA, 1989

[100] Sir Geoffrey Taylor, *Instability of liquid surfaces when accelerated in a direction perpendicular to their planes*, Proceedings of the Royal Society of London (Ser. A), 201, 1950

[101] Philip A. Thompson, **Compressible-Fluid Dynamics**, McGraw-Hill Book Comp., New York, 1972

[102] L. N. Trefethen, *Group velocity in finite difference schemes*, The Society of Industrial and Applied Mathematics Review, 24 (2), 1982

[103] Gretar Tryggvason and Salih Ozen Unverdi, *Computations of three-dimensional Rayleigh-Taylor instability*, Physics of Fluids (A), 2 (5), 1990

[104] Walter G. Vincenti and Charles H. Kruger Jr., **Introduction of Physical Gas Dynamics**, Robert E. Krieger Publishing Company, Malabar FL, 1975

[105] R. F. Warming and B. J. Hyett, *The modified equation approach to the stability and accuracy analysis of finite-difference methods*, Journal of Computational Physics, 14, 1974

[106] F. M. White, **Fluid Mechanics** (second edition,) McGraw Hill, New York, 1986

[107] Forman A. Williams, **Combustion Theory**, Addison Wesley, Redwood City CA, 1985

[108] V. Yakhot and S. Orszag, *Renormalized Group (RNG) methods for turbulent closure*, Journal of Scientific Computing, 1, 1986

[109] N. N. Yanenko, Z. I. Fedotova, L. A. Tusheva and Yu. I. Shokin, *Classification of difference schemes of gas dynamics by the method of differential approximation I, the one dimensional case*, Computers and Fluids, 11 (3), 1983

[110] N. N. Yanenko, Z. I. Fedotova, L. A. Kompaniets and Yu. I. Shokin, *Classification of difference schemes of gas dynamics by the method of differential approximation II, the two dimensional case*, Computers and Fluids, 12 (2), 1984

[111] Akira Yoshizawa, *Subgrid-scale modeling with a variable length scale*, Physics of Fluids (A), 1 (7), 1989

[112] D. Youngs, *Numerical simulation of turbulent mixing by Rayleigh-Taylor instability*, Physica D, 12, 1984

[113] D. Youngs, *Modelling turbulent mixing by Rayleigh-Taylor instability*, Physica D, 37, 1989

[114] David L. Youngs, *Three-dimensional numerical simulation of turbulent mixing by Rayleigh-Taylor instability*, Physics of Fluids (A), 3 (5), 1991

[115] Wolfgang Yourgrau and Stanley Mandelstam, **Variational Principles in Dynamical and Quantum Theory**, Dover, New York, 1979

[116] Ye Zhou, *Degrees of locality of energy transfer in the inertial range*, Physics of Fluids (A), 5 (5), 1993

[117] G. Zwas, *On two-step Lax-Wendroff methods in several dimensions*, Numerische Mathematik, 20, 1973

Technical Information Department • Lawrence Livermore National Laboratory
University of California • Livermore, California 94551